JUNIO 86 - 300 ptas.

TodoSpectrum

AÑO II - Número 22

REVISTA EXCLUSIVA PARA USUARIOS

PROBAMOS

Centronics, pero...¿cuál?

DESCUBRIMOS

Guía del hacker
Robin of the wood

CALCULAMOS

Números en coma flotante

INVESTIGAMOS

Las instrucciones ocultas del Z-80

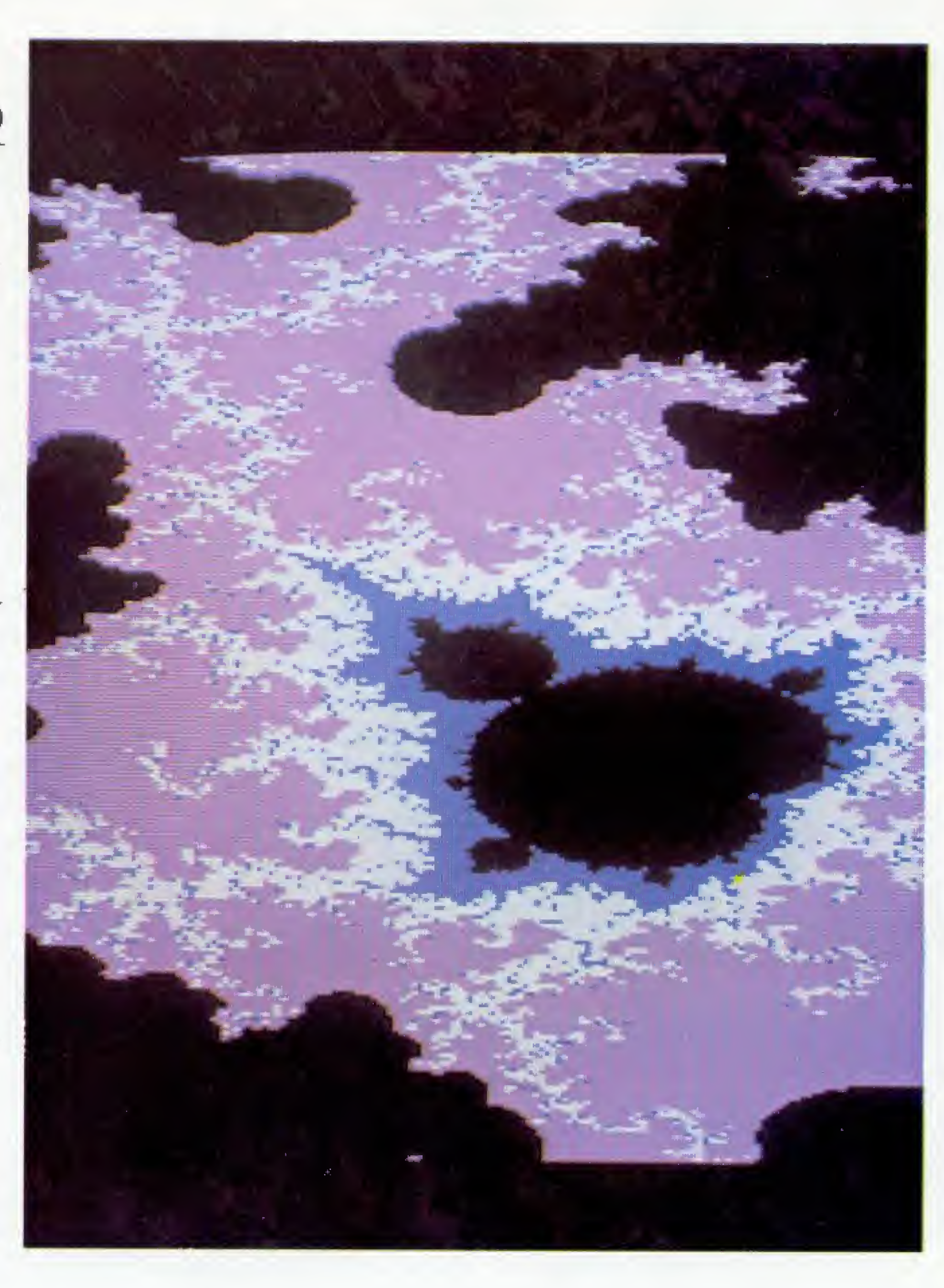


Suplemento QL

El conjunto de Mandelbrot

SUMARIO

AÑO II - N.º 22 - JUNIO 1986



SERVICIO DE EJEMPLARES ATRASADOS



Complete su colección de

Todospectrum

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.

Núm. 2 - 300 ptas.
Gráficos profesionales/Desplazamiento pixel a pixel/Utilización de rutinas/Construcción del interface centronics/Programas de utilidad para microdrive/Rutina reset en código máquina/Análisis del editor de textos Tasword/Interfaces para impresoras/Programas.

Núm. 3 - 300 ptas.
Novedades sonimag'84/Ampliando el Basic/Programas para ordenar programas/Gráficos con el VU-3D/Lenguaje Forth/Archivos en microdrive/Programación de un interface de impresora/Programas.

Núm. 4 - 300 ptas.
De profesión: programador/Consola para el Spectrum/Comparación código máquina-Basic/Análisis programa contabilidad/Calendario/Pascal/Programas.

Núm. 5 - 300 ptas.
Floppys para Spectrum/Diseño asistido por ordenador/64 Caracteres por línea/Juego de la vida/Pascal/Así hacemos las portadas/Control de evaluaciones/Programas.

Núm. 6 - 300 ptas.
Representación de funciones/Todos los caminos conducen a la ROM/Juegos/Pascal/Construcción de un lápiz óptico/Programas de gestión. El SITI/Logo: torgugas para todos/ Interrupciones del Z-80/Programas.

Núm. 7 - 300 ptas.
Del 48 al PLUS paso a paso/¿Plotter para Spectrum?/Juegos/Libros de código máquina/Lápiz óptico. Programación del montaje/El LOGO en la escuela/Pascal/Floppys para Spectrum/Programas.

Núm 8 - 300 ptas.
Amplía tu memoria... a 48 K/Arquitectura: análisis del PREYME/Juegos/FORTH. Nociones básicas/Una clave, please/QL Magazine. Ultimas novedades, análisis de software, Lenguajes/Aula informática con Spectrum/Programas.

Núm 9 - 300 ptas.
Spectrum parlanchín/Juegos/Aula informática con Spectrum/Análisis: Comercial 4/Pascal/Periféricos: Wafadrive/QL Magazine: EASEL lo mejor de PSION. Música con QL/Desplazamiento Pixel a Pixel, aportación de lectores/Programas/Programer II.

Núm. 10 - 300 ptas.
Discos: invesdisc 200/Juegos/Dos programas simultáneos/Protección del software/Conozca extremadura, consulte a su ordenador/Desensamblador Z-80/Sofware educativo/QL Magazine: novedades Informat, Hoja de cálculo, Ajedrez/Construya su propio Joystick/Pascal/programas.

DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE ZX
(sin necesidad de encuadernación)

Núm. 11 - 300 ptas.
Actualidad/La otra cara del LOGO/Juegos/El Spectrum habla castellano/SOFTAID ayuda para Etiopía/S.O.S. aquí el Spectrum/Dibujar con lápiz óptico/QL Magazine: Procesador de textos. Teclas de función programables/Programas.

Núm. 12 - 300 ptas.
Actualidad/Inteligencia artificial/Lápiz óptico dk'TRONICS/Juegos/Análisis/Bingo/Z-80 PIO/Código máquina/Análisis: MASTERFILE/Programas.

Núm. 13 - 300 ptas.
Actualidad/Discos: Discovery 1/Juegos/Inteligencia artificial/Un nuevo sistema operativo/QL Magazine: Archive, Cartridge doctor. Aplicaciones comerciales/Código máquina/Programas.

Núm. 14 - 300 ptas.
Actualidad, Spectrum 128/Cálculo de estructuras para ingenieros y arquitectos/HELP utilidades en microdrive/Juegos/El microdrive ese desconocido/Código máquina/QL Magazine: GRAPHIC QL. Juegos. Discos de 720 K/Un nuevo operativo/Programas.

Núm. 15 - 300 ptas.
Actualidad/Spectrum 128/Un nuevo operativo/Círculos redondos/Juegos/Utilidades: BETA-BASIC/QL Magazine: Introducción al SUPER BASIC. Nuevas utilidades/Hardware: Puertas lógicas/Código máquina/Programas.

Núm. 16 - 300 ptas.
Actualidad/Cinco horas con SCREEN$/Hardware práctico/Cálculos de infinita precisión/Juegos/Un nuevo operativo/QL Magazine: Gráficos en SUPER-BASIC. Dibujando con ratón. Archivos con Archive. Programa/La última batalla, Juego estratégico.

Núm. 17 - 300 ptas.
Actualidad/Gráficos interactivos/Juegos/Código máquina/Un nuevo operativo/Trucos de programación/QL Magazine: Radiografía del QL. Gráficos en SUPER-BASIC/Libros/Programmas.

Núm. 18 - 300 ptas.
Actualidad/Introducción al C/Libros/Juegos/De cinta a microcinta/Visión panorámica de los microprocesadores más comunes/QL Magazine: Copy de grises. Microprocesadores 68000, una familia numerosa/Curioseando en la ROM/Programas.

Para hacer su pedido, rellene este cupón HOY MISMO y envíelo a:

Todospectrum Bravo Murillo, 377
Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes ejemplares atrasados de TODOSPECTRUM .... al precio de 300 pts.

El importe lo abonaré
☐ POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO ☐ AMERICAN EXPRESS ☐ VISA ☐ INTERBANK

Número de mi tarjeta: ....

Fecha de caducidad .... Firma

NOMBRE ....

DIRECCION ....

CIUDAD .... C.P. ....

PROVINCIA ....

DIRECTOR:
Enrique F. Larreta
REDACTOR JEFE:
Emiliano Juárez
REDACCION:
Ignacio Borrell, Octavio López,
Antonio del Rio
DISEÑO:
Esteban Pérez

Editado por PUBLINFORMATICA, S. A.
Bravo Murillo, 377. 5.º A. Tel.:
733 74 13 - 28020 Madrid
Presidente:
Fernando Bolin
Director Editorial Revistas de Usuarios:
Juan Arencibia
Director de Ventas:
Antonio González
Producción: Miguel Onieva
Servicio al cliente:
Julia González. Tel.: 733 79 69
Administración:
PUBLINFORMATICA, S. A.
Publicidad Madrid:
Emilio García
Dirección, Publicidad y Administración:
Bravo Murillo, 377. 5.º A. Tel. 733 74 13.
Télex: 48877 OPZX e 28020 Madrid
Publicidad Barcelona:
Lidia Cendrós. Pelayo, 12. Tels. (93)
318 02 89 - 301 47 00, ext. 27 y 28.
08001 Barcelona
Depósito legal: M-29041-1984
Distribuye S.G.E.L. Avda. Valdelaparra,
s/n. Alcobendas (Madrid).
Fotomecánica: Karmat, C/ Pantoja, 10.
Madrid.
Fotocomposición: Espacio y Punto
Imprime: Héroes, C/ Torrelara, 8. Madrid.
Distribuidor en VENEZUELA,
SIPAM, S. A.
AVD. REPUBLICA DOMINICANA, EDIF.
FELTREC - OFICINA 4B BOLEITA SUR
CARACAS (VENEZUELA)
Esta publicación es miembro de la
Asociación de Revistas de
Información ari asociada a la
Federación Internacional de Prensa
Periódica, FIPP.

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia relacionada con suscripciones a:
TODOSPECTRUM EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5º
28002 MADRID
(Para todos los pagos reseñar solamentes TODOSPECTRUM)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377. 5.º A
Tel. 733 74 13 - 28020 MADRID

Si deseas colaborar en TODOSPECTRUM remite tus artículos o programas a Bravo Murillo, 377. 5.º A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados. A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.

EDITORIAL

CUANDO EL RIO SUENA...

Cuando apenas ha transcurrido un mes y medio desde la compra de Sinclair Research por Amstrad, las noticias sobre el futuro de los ordenadores Sinclair se suceden a la velocidad de la luz.

Sobre el QL, amenazado de muerte por Alan Sugar, se comenta en Gran Bretaña que un consorcio de empresas y grupos relacionados con este ordenador intenta adquirir los derechos de fabricación y los stocks existentes. Quanta, Digital Precision, Eidersoft y Timex podrían estar tras esta operación.

Y mientras el futuro del QL continúa en el aire, las noticias y rumores sobre el Spectrum son mucho más halagüeñas: Amstrad planea introducir una versión modificada a tiempo para la campaña navideña, durante la cual realizará un gran esfuerzo para apoyar la gama de productos Sinclair. Además, según nuestros colegas de la revista Sinclair User, está desarrollando un Super Spectrum que ofrecerá las mismas prestaciones gráficas y de sonido que el revolucionario Commodore Amiga, pero a un precio muy inferior. Al parecer estará basado en un microprocesador Z-80H (dos veces más rápido que el Z-80A) y tendrá un modo compatible Spectrum. Su memoria será de 128 Kbytes, pudiéndose ampliar hasta 1 Mbyte. Probablemente incorporará un cassette, pero el principal medio de almacenamiento de programas será la softcard. Se da por descontado que incluirá el sistema operativo CP/M, aceptando las mismas utilidades que los Amstrad CPC y PCW.

Se rumorea también que Amstrad podría lanzar con la marca Sinclair una consola de vídeojuegos similar a las Atari VCS, una impresora de matriz de puntos y una unidad de discos de 3 pulgadas para el Spectrum.

En España, tras el éxito de la 1.ª feria Amstrad, se especula con la posibilidad de que Indescomp se haga cargo de la distribución de los productos Sinclair y organice una feria Sinclair-Amstrad en septiembre.

Así pues, nos aguarda un verano muy movido, durante el cual se confirmará alguno de los proyectos comentados y se aclarará la estrategia de Amstrad respecto a la gama de ordenadores Sinclair.

TODOSPECTRUM

Las instrucciones o

No son instrucciones nuevas, siempre han estado ahí, pero no han sido mencionadas ni por ZILOG, ni por los autores que han escrito libros sobre el Z-80.

El procesador Z-80 de ZILOG permite a los programadores de código máquina trabajar con cerca de 700 instrucciones. De éstas, existe abundante documentación, tanto de la casa fabricante, ZILOG, como de otras fuentes, principalmente autores de divulgaciones y métodos de programación del ensamblador de este chip.

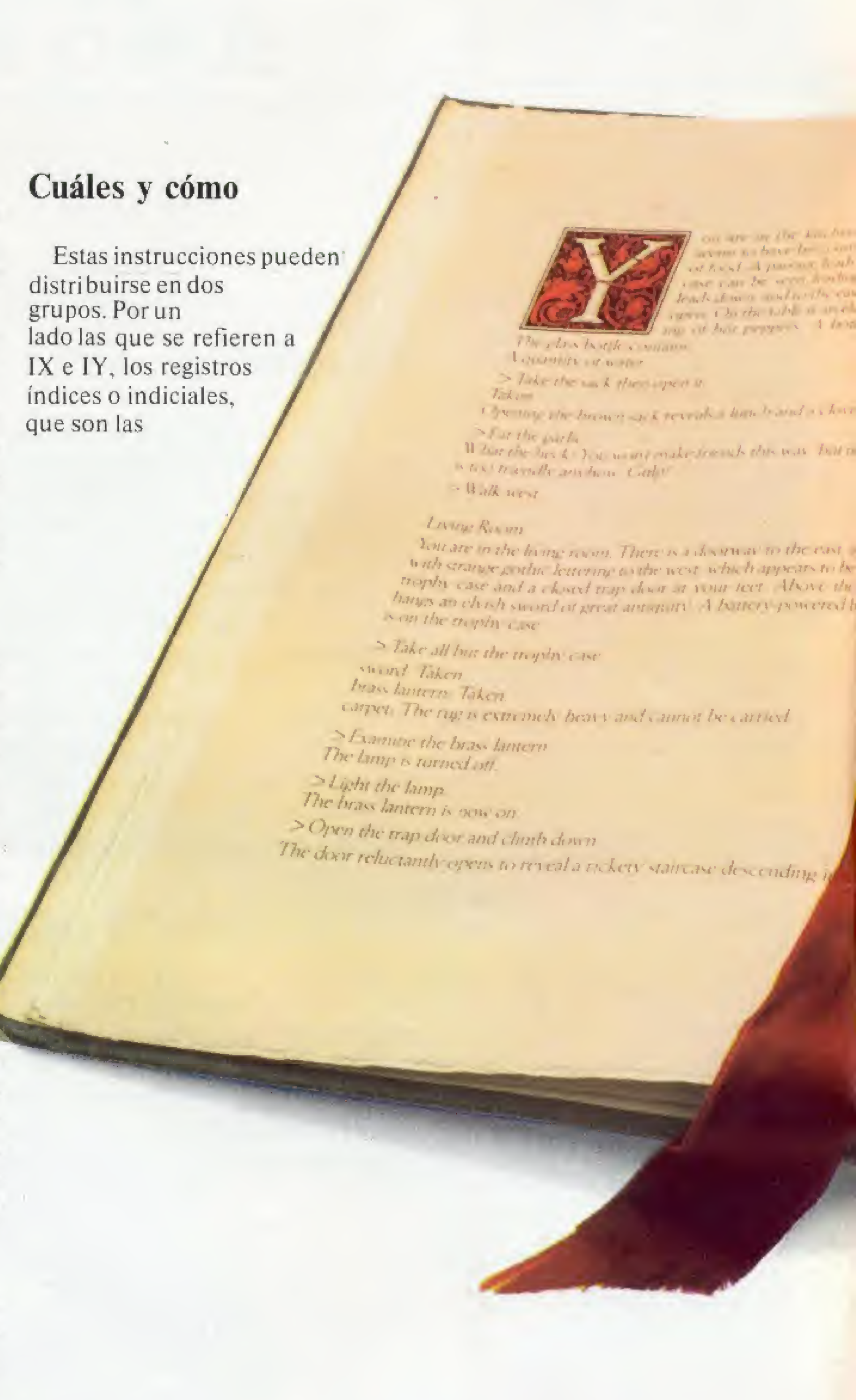
Sin embargo, existen unas 98 instrucciones ocultas que no aparecen en la información de ZILOG y de las que, seguramente, no has oído hablar nunca, pero que parecen funcionar en cualquier Z-80. Recientemente ha aumentado el interés por ellas, en parte debido a que algunos diseñadores de juegos las utilizan, bien sea por conveniencia, o bien para reforzar sistemas de protección que despisten por un rato a los «hackers».

Puesto que ZILOG no las referencia, la mayoría de los programas ensambladores y desensambladores no las reconocen. Así, no es fácil seguir la estructura de un programa ajeno en el que se utilicen, y desde luego, no podremos usarlas con sus nemónicos en un listado para ensamblar.

Resulta sorprendente que, de repente, aparezcan nuevas instrucciones de un procesador que ya tiene algunos años. Realmente estas instrucciones han estado siempre ahí, ZILOG no las menciona, pero ya se conocían. No es, por tanto, un descubrimiento sorprendente.

Cuáles y cómo

Estas instrucciones pueden distribuirse en dos grupos. Por un lado las que se refieren a IX e IY, los registros índices o indiciales, que son las



CULTAS del Z-80



más numerosas. El otro grupo contiene instrucciones referidas a desplazamientos, analicemos éstas primero.

Si nos fijamos en el conjunto de instrucciones de rotación y desplazamiento, veremos que para cada función existen 2 comandos. Uno realiza la función hacia la izquierda y el otro hacia la derecha (SLA-SRA, RL-RR, etc.). Sin embargo, existe una instrucción que no tiene su par, el desplazamiento lógico derecha, SRL, no tiene instrucción opuesta, que sería SLL (desplazamiento lógico izquierda). Esta rareza tiene confirmación en la secuencia numérica de códigos hexadecimales de las instrucciones. Entre CB30 y CB37 hay un intervalo en el que los códigos no tienen asociada ninguna instrucción. Precisamente donde debería estar el grupo de SLL.

La mayoría de los ensambladores y monitores no reconocen las 98 instrucciones no referenciadas por Zilog

Y funcionan

Podemos operar con estos códigos y veremos que las instrucciones son aceptadas por el procesador, y que realmente ocurre un deplazamiento lógico a la izquierda, o algo que se le parece. Mientras SRL desplaza el registro a la derecha en un bit, el contenido del bit 0 pasa al banderín de acarreo y el bit 7 se llena con un 0, SLL no hace lo simétrico. El registro si es desplazado a la izquierda, el bit 7 se copia en el bit de acarreo, pero en vez de un 0, un 1 es colocado en el bit 0. SLL afecta al registro F exactamente como lo haría su gemela SRL.

Todas las variantes de esta instrucción se encuentran en la Tabla 1, con el resto de comandos ocultos.

Más instrucciones

El otro grupo, más numeroso, de instrucciones «reaparecidas» se refieren en su totalidad a operaciones con los registros índice IX e IY. Observando una lista de instrucciones del Z-80, podemos apreciar cómo existe una gran relación entre los códigos de las instrucciones que operan con el registro HL, y las que lo hacen con IX e IY. Por ejemplo, tomemos LD (HL),A, cuyo código es 77. La operación similar con IX será: LD (IX+des),A; su código es DD 77 nn. Como veremos el código de la instrucción de IX es el mismo, con DD delante y añadiendo un octeto que indicará el desplazamiento sobre el índice. Similarmente, para IY tenemos que hay que preceder el código de la instrucción de HL por FD, todo acompañado del inevitable octeto del desplazamiento (LD (IY+des),A = FD 77 nn).

Investigando

Si utilizamos el mismo método de búsqueda que con SLL, encontraremos bastantes huecos en la secuencia de los números de los códigos de las instrucciones referidas a los índices. Tomemos SBC A,H (9C Hex), para IX e IY no hay equivalente. Formemos el código DD 9C que correspondería, según la regla de equivalencia, a la instrucción similar de IX. Si operamos con este código obtendremos que el contenido del octeto superior de IX será restado de A con acarreo. Indénticamente sucede con AND H (A4 Hex), DD A4 realizará un AND lógico entre el acumulador y la parte alta de IX. Con FD R4 obtendremos los mismos resultados, pero ahora con la parte alta del registro IY.

Nuevos registros

Todas las instrucciones ocultas restantes de este grupo trabajan de esta forma. Todas operan con los octetos altos o con los bajos de los registros índices. Y todo idéntico a las instrucciones que manejan H y L como registros independientes. Sus códigos serán los mismos, precedidos por DD si estamos utilizando alguno de los octetos de IX, o por FD

Estas instrucciones se distribuyen en dos grupos, las que se refieren a los registros índice, que son las más numerosas, y las de desplazamiento

si son las partes del registro IY las referenciadas. Para identificar estos octetos que forman parte de IX e IY los llamaremos XH (octeto alto de IX) y XL (octeto bajo de IX), análogamente YH e YL serán las mitades alta y baja de IY, respectivamente (H por high y L por low, es inevitable usar el inglés en este caso).

Como es de esperar, debido a la similitud entre instrucciones, los cambios en los banderines, ocasionados por estos comandos serán los mismos que los forzados por las ins-

trucciones que trabajan con H o L. La lista de comandos completa está en el cuadro 1.

Nada espectacular

¿Qué ventajas aportan estas instrucciones «nuevas»? Como ya dijimos pueden ser utilizadas para despistar las incursiones de «hackers» dispuestos a bucear en nuestros programas, ya que no son aceptadas/reconocidas por casi ningún ensamblador/desensamblador. También podemos utilizarlas para disponer de XH, XL, YH e YL como registros de 8 bits adicionales.

Está claro que estas instrucciones no aportan soluciones espectaculares a la programación en código máquina, tomémoslas como lo que son y dispondremos de 98 comandos más que pueden servirnos en algún momento.

Joaquín Mateos

Tabla 1: Lista de comandos ocultos del Z-80

SLL(HL)	CB 36	SLL C	CB 31
SLL(IX+nn)	DD CB nn 36	SLL D	CB 32
SLL(IY+nn)	FD CB nn 36	SLL E	CB 33
SLL A	CB 37	SLL H	CB 34
SLL B	CB 30	SLL L	CB 35
ADC A,XH	DD 8C	LD XH,A	DD 67
ADC A,XL	DD 8D	LD XH,B	DD 60
ADD A,XH	DD 84	LD XH,C	DD 61
ADD A,XL	DD 85	LD XH,D	DD 62
AND XH	DD A4	LD XH,E	DD 63
AND XL	DD A5	LD XL,A	DD 6F
CP XH	DD BC	LD XL,B	DD 68
CP XL	DD BD	LD XL,C	DD 69
DEC XH	DD 25	LD XL,D	DD 6A
DEC XL	DD 2D	LD XL,E	DD 6B
INC XH	DD 24	LD XL,XH	DD 6C
INC XL	DD 2C	LD XL,nn	DD 2E nn
LD A,XH	DD 7C	LD XH,XL	DD 65
LD A,XL	DD 7D	LD XH,nn	DD 26 nn
LD B,XH	DD 44	OR XH	DD B4
LD B,XL	DD 45	OR XL	DD B5
LD C,XH	DD 4C	SBC A,XH	DD 9C
LD C,XL	DD 4D	SBC A,XL	DD 9D
LD D,XH	DD 54	SUB XH	DD 94
LD D,XL	DD 55	SUB XL	DD 95
LD E,XH	DD 5C	XOR XH	DD AC
LD E,XL	DD 5D	XOR XL	DD AD

Para obtener los comandos referidos a IY, sustituir DD por FD en los códigos de instrucción.

ROBIN

Hoy es una de esas tardes lluviosas de primavera en las que lo mejor que podemos hacer es sentarnos delante del ordenador para dedicarnos a nuestra actividad favorita, modificar programas. Esta vez se trata de ROBIN OF THE WOOD. Es un juego al estilo de SABRE WULF en el que encarnamos al famoso héroe de los bosques de Sherwood. Nuestra misión consiste en recuperar la flecha de plata. Para ello debemos llevar al anciano ENT (árbol con aparicneica humana) el dinero que le quitemos al obispo de Peterborough. A cambio nos irá entregando nuestras armas y tres flechas mágicas con las que podremos entrar en el castillo y cumplir nuestra misión.

El número de pantallas parece ser muy elevado aunque las podemos recorrer rápidamente sin excesivas dificultades. Los gráficos y el sonido están muy bien realizados.

Para empezar, tenemos que ver cómo se carga para poder acceder al código máquina. Lo primero que encontramos es un pequeño cargador en BASIC que se puede parar sin muchos problemas. Contiene una línea REM con un programa código máquina que carga de la cinta un enorme bloque sin cabecera, empieza en #4000 y tiene #BFFO bytes.

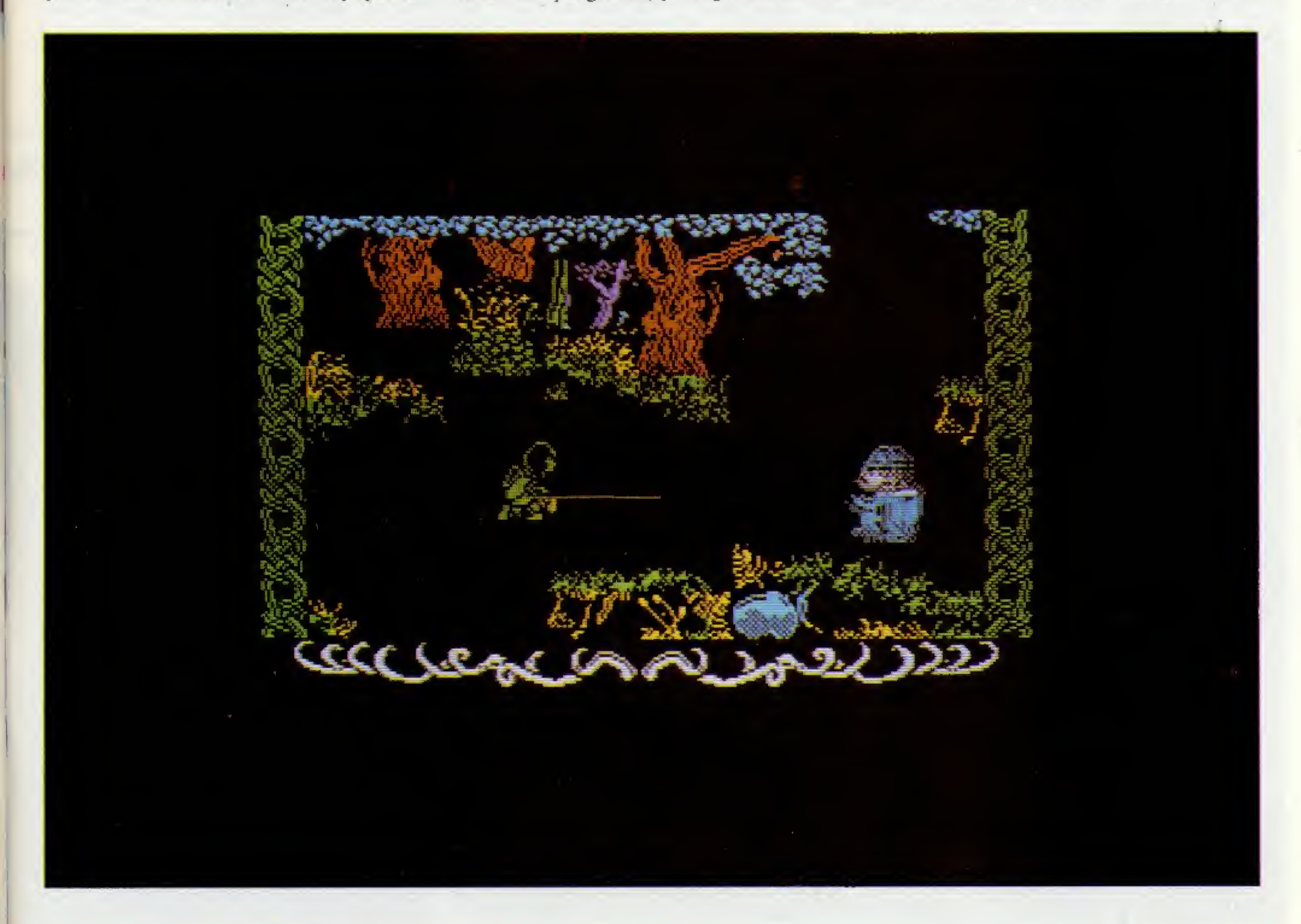


Es posible que tengamos bastantes problemas para manejarlo. La primera idea consiste en cargar sólo el principio del programa e intentar ver qué había en las variables del sistema BASIC cuando se grabó el programa. El intento no tiene mucho éxito. Lo que aparece en esta zona no tiene ningún parecido con las variables BASIC. Deben de estar enmascaradas o corresponder con algunas tablas o gráficos. Tendremos que tratar de separar el bloque correspondiente al final del programa para ver qué es lo que hace cuando se lanza en #BF70. Para ello hacemos un cargador que lo coloca a partir de #3FF0, coloca el RAMTOP en #6FFF y hace un NEW (salto a #11b7). De esta forma conseguiremoss tener en memoria los últimos 32k del programa. Esto no es suficiente para que funcione pero nos sirve para comenzar a investigar.

El hecho de que el bloque principal sea tan largo plantea serios problemas si

N OF THE WOOD

queremos escribir al final un cargador que incluya algunas opciones nuevas por varias razones. Primero hay que localizar un sitio en memoria donde colocarlo sin que afecte a la ejecución del resto del programa, y luego buscar algún método para cargar todo el bloque con nuestro programa en memoria. Analizando un poco más lo que hay en



PROGRAMA 2

```
  10 LET n=61440: RESTORE
  30 FOR i=1000 TO 1090 STEP 10
  40 READ a$,a: LET s=0
  50 FOR j=1 TO LEN a$-1 STEP 2
  60 LET d=16*(CODE a$(j)-48-7*(
a$(j)>"9"))+CODE a$(j+1)-48-7*(a
$(j+1)>"9")
  70 POKE n,d: LET n=n+1: LET s=
s+d: NEXT j
  80 IF s<>a THEN  PRINT "Error
en la linea ";i: STOP
  90 NEXT i
 100 RANDOMIZE USR 61440
1000 DATA "310000DD2100401116A03
EFF37CD56053EC1BC20EE213CE022EDC
D2131F01116",3101
1010 DATA "E0010E01EDB0CD24D03E4
4CD32D0C379BFE17E23E5FEFFC83222E
12120E1CDE1",4555
1020 DATA "CF3A20E1C6043220E118E
63A21E1C610672E202220E1C93E45CD3
FD0CD81CE21",3668
1030 DATA "20202220E1CD16E0494E4
D4F5254414CFFCDEFE03E353001AF323
2BECD16E053",3506
1040 DATA "494E2044494E45524FFFC
DEFE03ED83001AF3230DFCD16E04F424
953504F2046",3391
1050 DATA "494A4FFFCDEFE0113E003
80311EDE37B3226B97A3242B9CD16E04
45255494441",3479
1060 DATA "204255454E4FFFCDEFE03
ED83001AF32BBBECD16E053494E204A4
142414C4945",3466
1070 DATA "53FFCDEFE03EC8CE0032F
EDBCD16E053494E2053484552494646F
FCDEFE03ED3",4429
1080 DATA "30023E3832BDCFC367D0A
FDBFE2FE61F20F83E7FDBFE2FE608201
43EFDDBFEE6",4373
1090 DATA "0220EFCD16E0205349FFC
D30E037C9CD16E0204E4FFFCD30E0A7C
92F203000",3767
```

la cinta se descubre que el bloque tiene exactamente 48k, aunque los últimos 16 bytes no se cargan para dejar un poco de sitio a la pila. Esto nos da una primera esperanza. Quizás los últimos bytes del programa no sirvan para nada y podamos no cargarlos, con lo que tendríamos un poco de sitio para manipular toda la memoria.

Pasamos a ver qué es lo que ocurre cuando se lanza el programa en #BF70. Nos encontramos con un bucle que se entiende hasta #C055 y que se cierra sobre #BF81. Tiene que ser el bucle principal y lo que hay antes de esta última dirección corresponde con el menú del principio. Este tiene dos puntos de entrada, uno en #CD6C que se usa la primera vez, y otro en #CD89 que se usa todas las demás. La diferencia entre estas dos entradas está en que en la primera hay una llamada a #8CAA que corresponde a la frase digitalizada que dice al principio y borra la pantalla. Las tablas de esta frase están a partir de la dirección #E500 y son destruidas nada más utilizadas. Detrás de aquí no parece haber nada importante. Probamos a cargar solamente hasta esa dirección y el juego funciona perfectamente. Esto nos da un margen importante para poder trabajar. Instalamos en un microdrive dos copias del programa, una es la que contiene únicamente los últimos 32k, donde está todo el programa, el resto son tablas y gráficos. La otra tiene todo el programa sin la pantalla de presentación colocada en una dirección distinta preparada para probar los POKEs que encontremos. Trataremos de ir analizando paso a paso todo lo que hace desde que se lanza el programa, procurando preocuparnos más por comprender qué es lo que está haciendo que por descubrir algún POKE. Lo hacemos así, no solamente para llegar a un conocimiento más profundo del programa sino también por que para probar un POKE tenemos que destruir la copia actualmente en memoria y cargar la otra, lo cual requiere un tiempo considerable.

En la dirección #CD89 empieza la rutina que gestiona el menú. Lo primero que hace es llamar a la dirección #CFEE. Esta es una curiosa rutina que lo que es ir colocando a partir de la dirección #FD00 una tabla de 256 bytes en la que cada uno es el número de orden dado la vuelta bit a bit. Así en la dirección #FD01 coloca el dato #80 (#01 al revés). La utilidad de esta tabla debe ser para a partir de un gráfico (Robin mirando hacia la derecha) generar de forma rápida su simétrica (Robin mirando hacia la izquierda). De esta forma se ahorran la mitad de los gráficos correspondientes a Robin, los soldados, el druida y el obispo. También crea otra tabla con las direcciones de pantalla en función de la línea de que se trata. Con esto consigue una mayor velocidad al no tener que calcular la dirección cada vez que quiere pintar un gráfico.

Siguiendo con las rutinas a las que

El largo bloque principal plan
con nuevas opciones.



llamas en esta zona nos encontramos con que en la dirección #D095 selecciona el modo de interrupción IM2. La dirección de la rutina de interrupción es # DFE7 y su única misión es encargarse de parte del sonido.

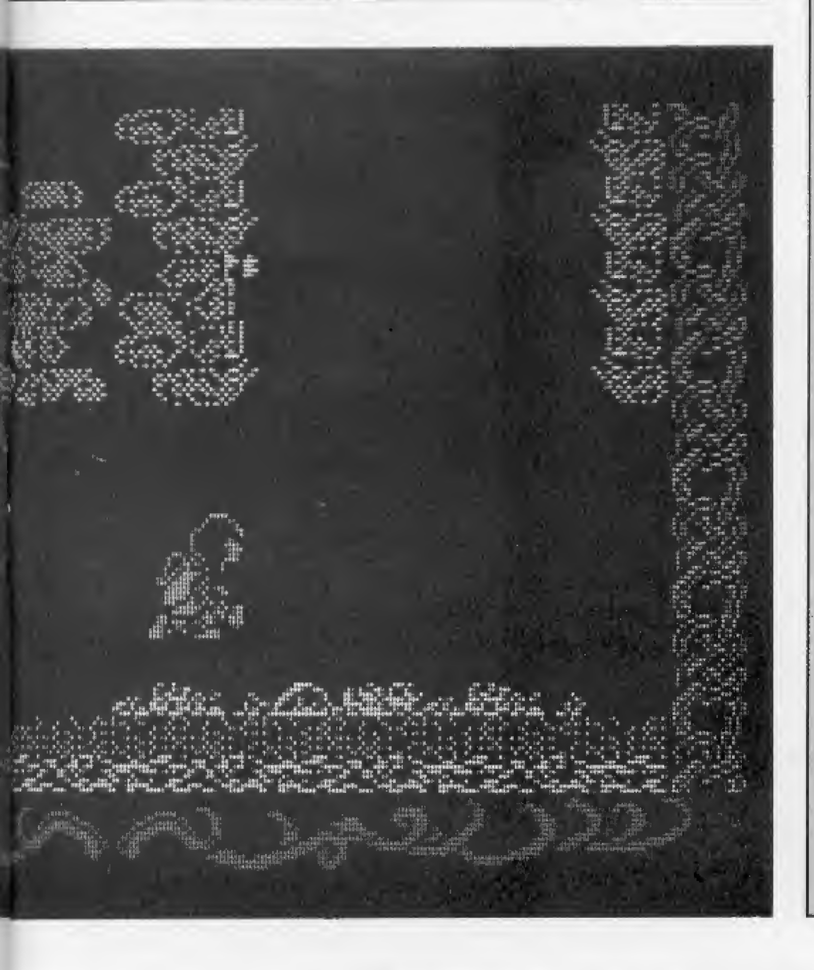
Por lo demás aquí nos encontramos con un motón de llamadas encargadas del menú. Borran la pantalla o la copia de pantalla, escribir las opciones, pintar la espada, etc. La forma en que se manejan las rutinas de impresión de texto en pantalla es excesivamente rígida e incómoda de usar, por lo que en el cargador final tendremos que incluir una que nos facilite esa labor.

Un poco más adelante nos encontramos con que testea el teclado, y si está pulsado el 0 salta a la dirección #CDEC. Aquí ya empieza a inicializar variables. En la dirección #BF6F coloca un sospechoso #OF. Se trata del dato que controla el estado de salud de nuestro héroe. Buscando en qué punto se modifica logramos el POKE de la inmortalidad, basta con NO pear la dirección #BE32.

Hay una serie de variables que se inicializan eligiendo aleatoriamente entre una serie de posibles valores. Tienen que ser los correspondientes a los elementos que cambian de una partida a otra, esto es, la posición de comienzo, la de ENT, la del sheriff de Nothingam y la del druida. Trataremos de identificar cada uno de estos. Para ello echamos un vistazo a cuando se utilizan estas variables y nos encontramos con que en varias ocasiones se compara alguna de ellas con el contenido de #C548, que también se inicializa aleatoriamente. Esto nos lleva a la conclusión de que esta última contiene el número de la pantalla en que nos encontramos. Este número se guarda sobre dos bytes, lo que parece indicar que tiene un mapa de enormes dimensiones. Para cada pantalla hay un byte con el cual ya accede a la definición de la pantalla. De esta forma el número de pantallas definidas en memoria es bastante menor ya que en el mapa existen muchas iguales. Para saber a qué corresponde cada una de las demás variables pokeamos el punto en que compara con la pantalla en que nos encontramos. Haciendo esto llegamos a las siguientes conclusiones: en #D3B2 se guarda la pantalla de ENT, en #D3B4 la de la cabaña del druida y en #D3B8 la del sheriff.

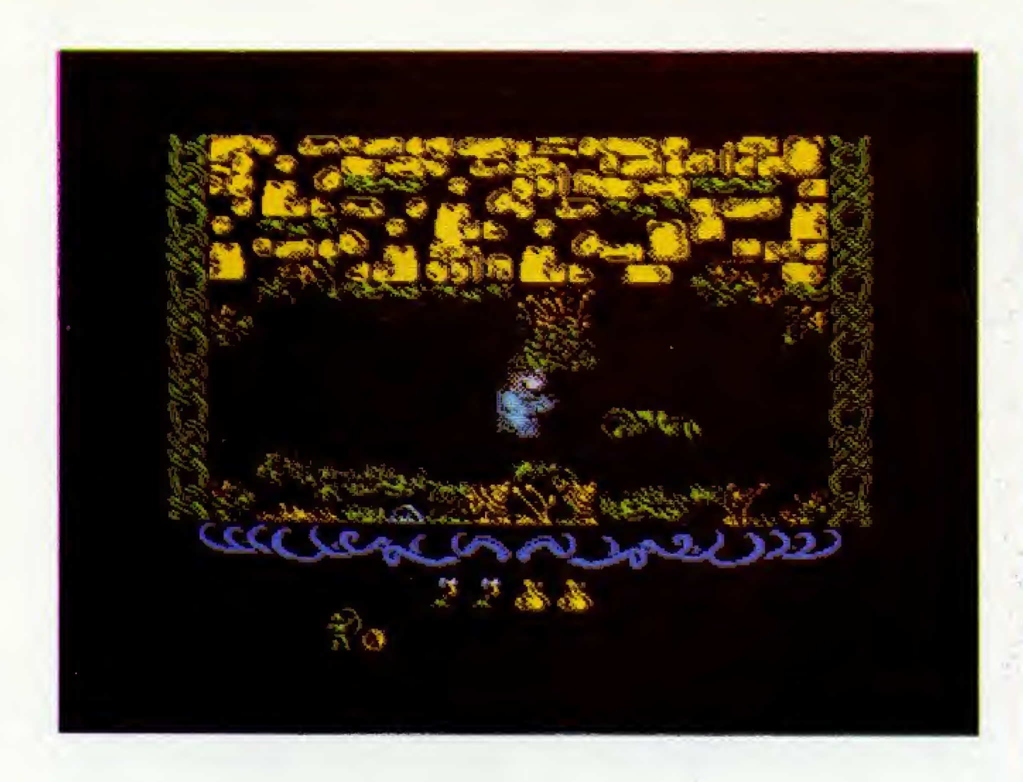
Si queremos ver el mapa completo tendremos que hacer un programa que nos permita movernos rápidamente de una habitación a otra. El interferir el bucle principal para que en función de la tecla que pulsemos nos movamos en distintas direcciones por el mapa no tiene mucha dificultad. El que escriba el número de pantalla presenta algunos problemas, pero finalmente funciona y lo tenéis en el programa 1. Para utilizarlo tenéis que ejecutar el programa, esperar un poco a que pokee y luego cargar la copia original. Cuando acabe el juego empezará normalmente, pero si pulsáis la tecla O se detendrá y podréis pasar a cualquier pantalla pulsando las teclas 1 a 4. También se ha incluido el POKE de inmortalidad y otro que permite conse-

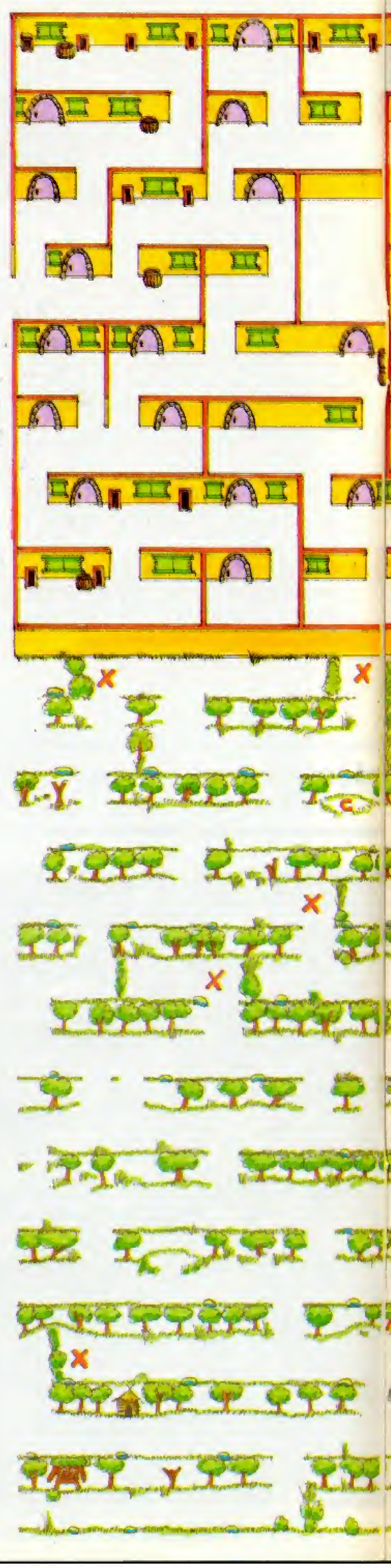
...eá problemas al intentar escribir un cargador



PROGRAMA 1

```
   10 LET n=61440: RESTORE
   30 FOR i=1000 TO 1040 STEP 10
   40 READ a$,a: LET s=0
   50 FOR j=1 TO LEN a$-1 STEP 2
   60 LET d=16*(CODE a$(j)-48-7*(
a$(j)>"9"))+CODE a$(j+1)-48-7*(a
$(j+1)>"9")
   70 POKE n,d: LET n=n+1: LET s=
s+d: NEXT j
   80 IF s<>a THEN  PRINT "Error
en la linea ";i: STOP
   90 NEXT i
  100 RANDOMIZE USR 61440
 1000 DATA "310000DD2100401100A53
EFF37CD56053EEBBC20EE2116E02282B
F2138F01116",2974
 1010 DATA "E016600EDB0AF3230DF3
232BECD24D03E44CD32D0C379BF2A48C
57DE60FFE0A",4020
 1020 DATA "3802C607C630327AE0253
E202023E313277E07D1F1F1FE60FC
630FE3A3802",2684
 1030 DATA "C6073278E02175E03E603
2F2D7CD19D63EEFDBFE1F3A74E0380A2
FF53EEFDBFE",4417
 1040 DATA "1F30F9F13274E03CC23BC
53EF7DBFEE60FE15FC3EDBF0070A0303
02030",3887
```





guir las armas sin necesidad de ninguna bolsa de dinero.

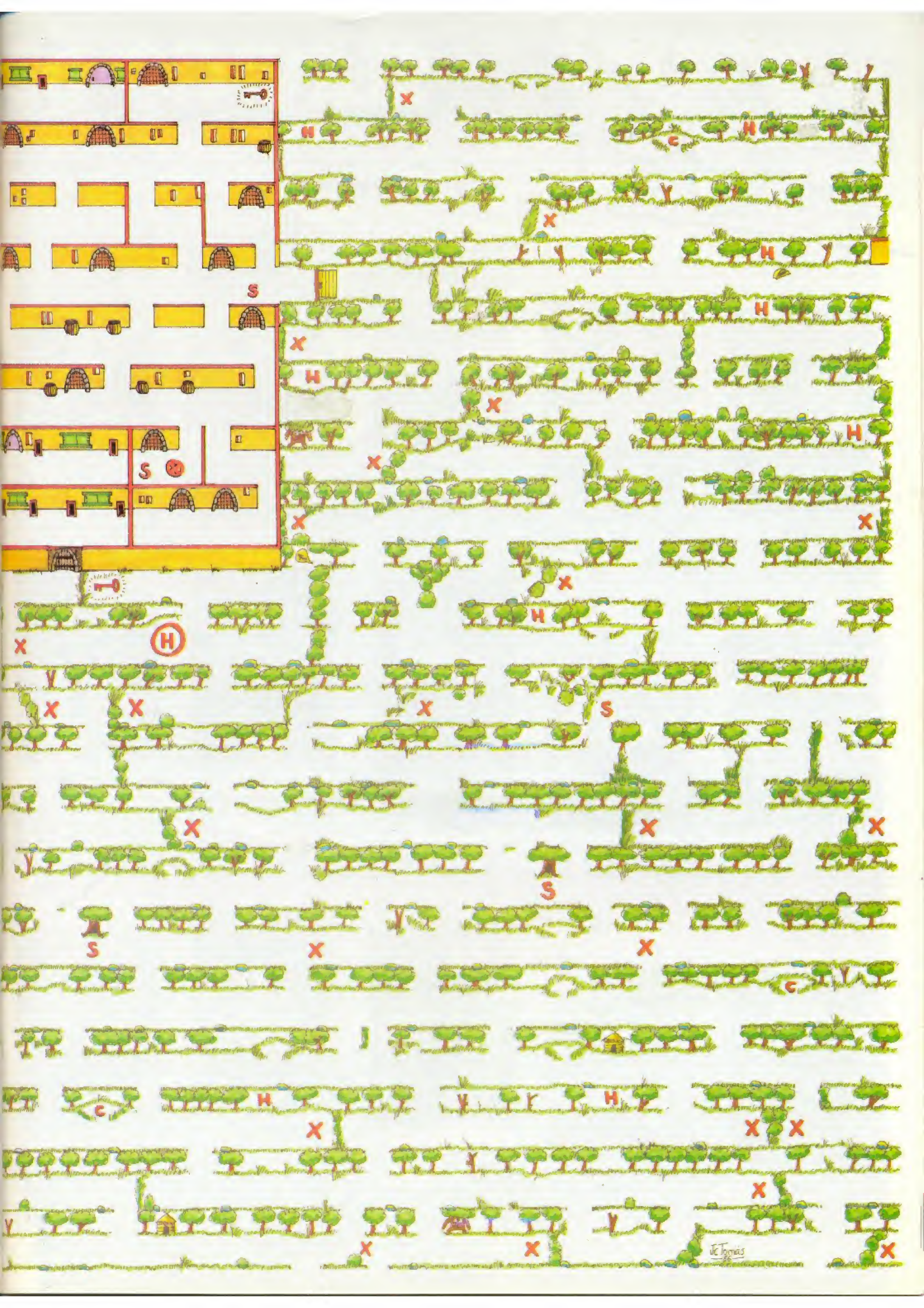
Si seguimos investigando nos encontramos con que a partir de la dirección # D4AD está la tabla de los objetos repartidos por el mapa. Para cada objeto hay cinco bytes, de los que los dos primeros indican en que pantalla está, el tercero y cuarto la posición dentro de la pantalla y el quinto el tipo de objeto de que se trata. A partir de #D595 están los códigos de las 8 cosas que llevamos en ese momento. Buscando cuando se utiliza descubrimos la rutina de la dirección # DF03, que es llamada directamente desde el bucle principal. Se encarga de comprobar si estamos en la habitación de ENT, si llevamos el dinero y de darnos las armas. NO peando la dirección # DF30 se consigue que no sean necesarias las bolsas de dinero.

Buscando en las rutinas que se llaman desde el bucle principal encontramos los demás POKEs que se han incluido en el programa 2. Es un cargador normal que nos pregunta por los POKEs después del menú de opciones.

Nuestro análisis termina aquí, a la 1.30 de la noche, pero quedan muchas cosas por descubrir. Quizás alguien se anime a buscarlas.

Manuel Arana







SERVICIO DE

Completa tu colección de ZX.
A continuación te resumimos el contenido de los ejemplares atrasados en existencia.



Núm. 3/300 ptas.
El Spectrum por dentro. Quince programas, juegos y montajes Software.



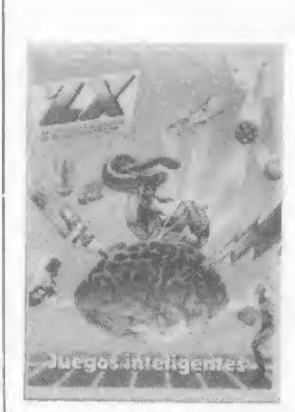
Núm. 4/300 ptas.
QL, el nuevo Sinclair. Dieciocho programas, juegos, montajes, ideas/Novedades.



Núm. 5/300 ptas.
Gráficos y sonido en el Spectrum/Libros/Software/ 13 programas.



Núm. 6/300 ptas.
Construya su propio juego/13 programas y montajes/ideas/Software.



Núm. 7/300 ptas.
Juegos inteligentes/Software/ 11 programas/Libros.



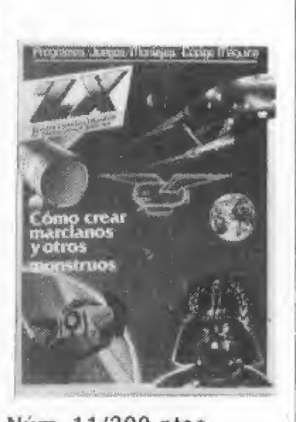
Núm. 8/300 ptas.
La aventura es la aventura/12 programas/Juegos y montajes/Código máquina.



Núm. 9/300 ptas.
Construye tu propio juego. Catorce programas para el verano. Gráficos en el Spectrum.



Núm. 10/300 ptas.
Catorce programas educativos: geografía, cramer, gráficos, razones trigonométricas, elongación. Código máquina.



Núm. 11/300 ptas.
Cómo crear marcianos y otros monstruos.
Diez programas satélites de júpiter, rescate, interés, círculo, préstamo hipotecario.



Núm. 12/300 ptas.
Presentación del Spectrum Plus. Forth, capítulo 1. Gráficos en el Spectrum, 4 parte. Libros. Programas y montajes.



Núm. 13/300 ptas.
Guia del software para el Spectrum todos los programas del mercado. Forth, capitulo 2. Visitamos Sinclair Research. Libros. Programas.



Núm. 14/300 ptas.
Cómo jugar al Hobbit. Gráficos de funciones. Programas de ajedrez. Conexiones con el P I/O. Programas Multiplic, enseñar deleitando. Libros, Forth, tercera parte.



Núm. 15/300 ptas.
Simuladores de vuelo. Forth, cuarta parte. Montajes: Reloj digital para Spectrum. BASIC para principiantes. Libros. Programas.



Núm. 16/300 ptas.
Cassettes: solución a los problemas de grabación. Test de Psicología. Sistema de Desarrollo para el ZX-81. Cinemática. Programas. Animación Gráfica. BASIC para principiantes (2). Forth, quinta parte.



Núm. 17/300 ptas.
Mapa de Atic-Atac. Estira de caracteres. Dinámica de una partícula. Libros. QL Magazine. Programas. Convertidor analógico-digital con el P I/O.

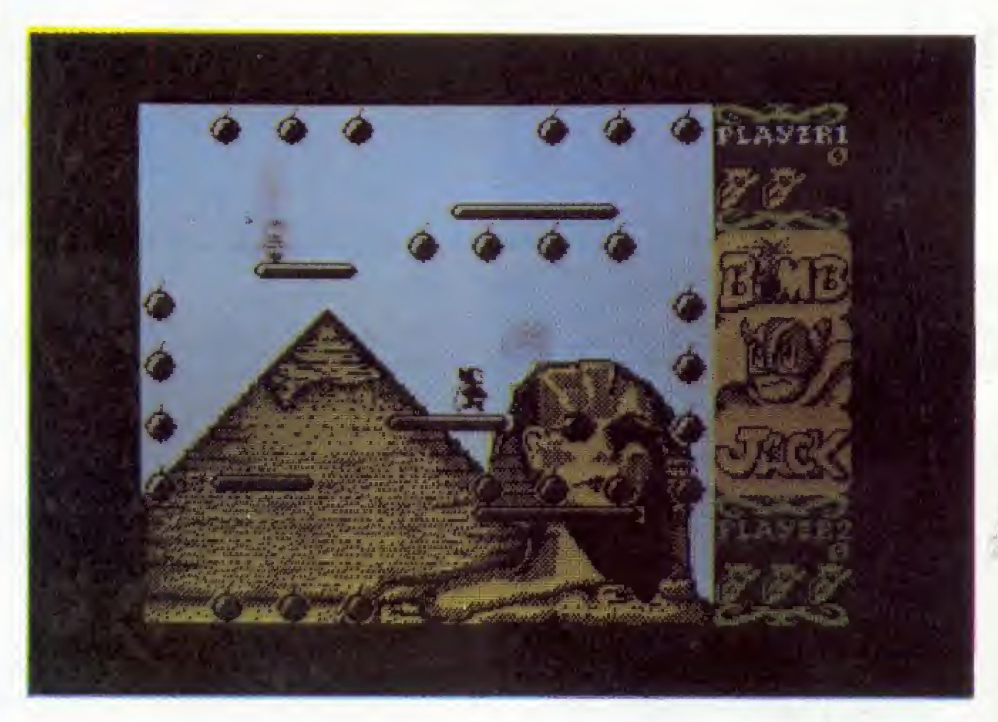
BOMB JACK

ELITE

SPECTRUM 48 K

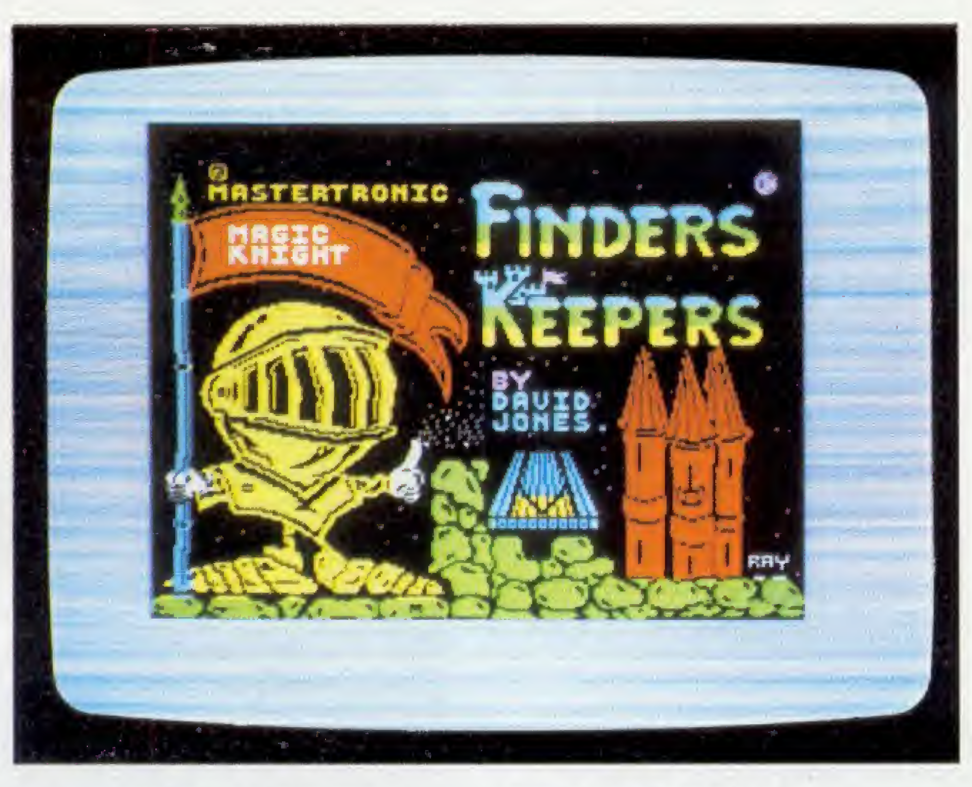


Cierta organización terrorista ha sembrado de bombas ciertos monumentos de todo el mundo. Nadie se atreve a acercarse, pues en cuanto una de ellas sea tocada comenzará una reacción en cadena que hará que todas las demás se vayan activando. Sólo un ser que volara a la velocidad del rayo podria triunfar en una misión que ha hecho temblar a los más osados expertos del mundo en este tipo de operaciones.
De pronto:
zzzZZZCHHOWSSSsss ¿un pájaro? ¿un avión?, ¡no! es Super-ratón que llega dispuesto a librar al mundo de la plaga que intenta acabar con el equilibrio y la paz. Debes ponerte en su Superpellejo, y, aprovechando sus Superpoderes, intentar desactivar el mayor número de bombas posible. Eso sí, tendrás que esquivar a la serie de «personajillos» que intentarán evitar que lo consigas, son de la más ruín calaña del planeta y han conseguido la ratonita, única substancia capaz de vencer a tus Superpoderes y que acabará con una de tus cinco Supervidas en cuanto te toque.
En todo caso lo que hace atractivo a este juego no es tanto la parte gráfica como la rapidez y suavidad de movimientos del protagonista, que hace que, a poco que lo controle, el usuario se sienta todo un Superusuario dando botes de un extremo al otro de la pantalla.
El mayor cúmulo de puntos lo lograremos si desactivamos las bombas en el orden correcto, es decir, primero las que estén encendidas. Hay que prestar atencióna los dicos que aparecen de tanto en tanto por el extremo superior derecho de la pantalla; la «P» inmoviliza unos segundos a nuestros enemigos y nos permite convertirnos en atacantes, la «E» da una vida extra y la «B» da puntos.
Entretenido y tremendamente adictivo.



FINDERS KEEPERS

DRO SOFT
SPECTRUM 48 K



Dro es la firma encargada de la distribución en España de los productos de Mastertronics, casa inglesa que comenzó la guerra de los precios y a la que parece que le ha ido bastante bien allá por el Reino Unido. Aparecen en España sus cintas a un precio algo por encima de las 700 ptas.; todo un logro en juegos que, sin llegar al nivel de calidad de los últimos grandes éxitos, no están nada mal. Pertenece Finders Keepers a la categoría de los juegos de arcade-aventura, pues no sólo deberemos ser diestros en el manejo del joystick, sino que también habremos de recorrer metódicamente los laberintos recogiendo los objetos más interesantes y comerciando para conseguir reunir un buen tesoro para nuestro rey.

Y es que el argumento tampoco está falto de originalidad: resulta que llega el día del cumpleaños de la hermosa princesa del reino de Ibisima y el rey no ha encontrado ningún obsequio que pueda satisfacerla.



Desesperado ha recurrido a tí, caballero mágico del reino, para que busques en el temido laberinto perdido los famosos tesosros que en él se hallan.

Consta el juego de tres fases distintas: primero aparece el protagonista en el palacio junto al rey, desde donde debe ser teletransportado a las cercanías del laberinto. Una vez allí ha de pasar varias pantallas al estilo del mítico Manic Miner para llegar a la parte donde cambian por completo la forma de moverse del personaje y el cambio entre pantallas.

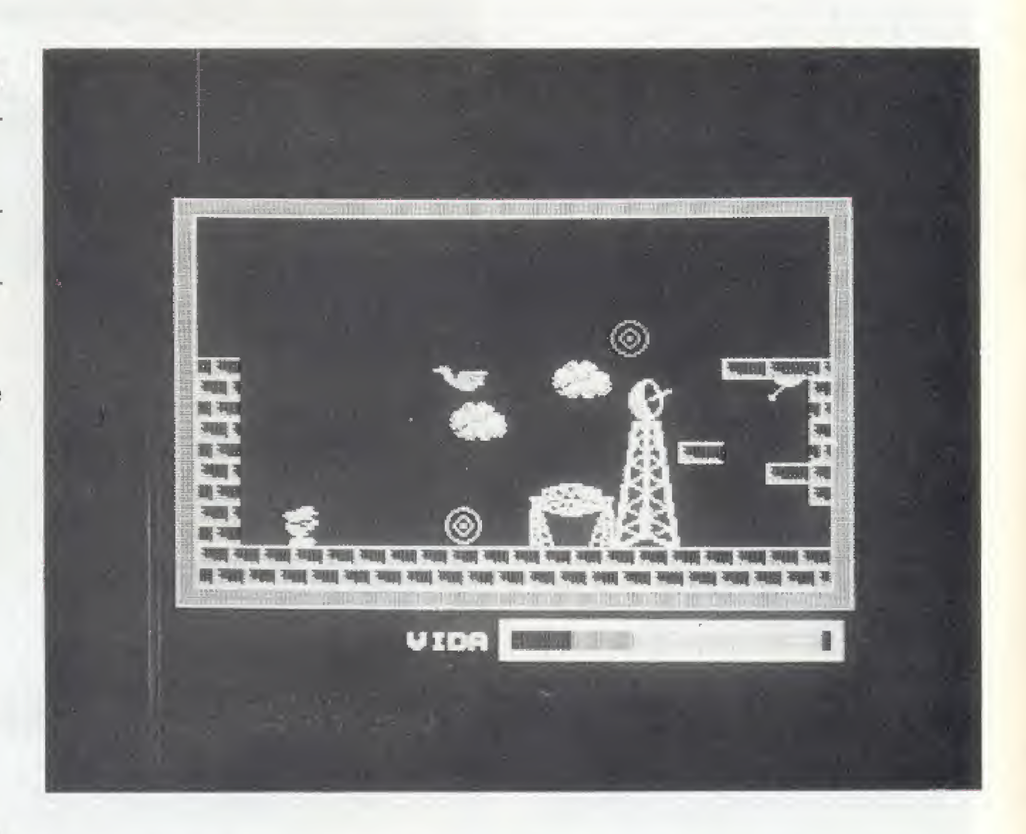
Son estos cambios en la estructura del programa, junto al buen gusto demostrado en la parte gráfica, las principales virtudes del juego, que se muestra muy completo también en otros detalles, especialmente los que atañen a la presentación al comienzo del mismo.

Nos encontramos, pues, ante una cinta muy interesante por su excelente relación calidad-precio, no es por nada por lo que ha conseguido ser «number one» en ventas en las listas inglesas del pasado año.

PHANTOMAS

DINAMIC

SPECTRUM 48 K



Todo comenzó en el año 2077. El programa para desarrollo de mutantes de la base científica de Gurnar había sido un éxito y el ordenador para la reproducción masiva creó el primer Phantomas, un mutante especializado en el robo, en el saqueo y en el pillaje. Este ser, dotado de una inteligencia privilegiada para eludir su captura por el enemigo, iba a ser utilizada en la batalla sideral de las lunas rojas de Alfa-Centauri.

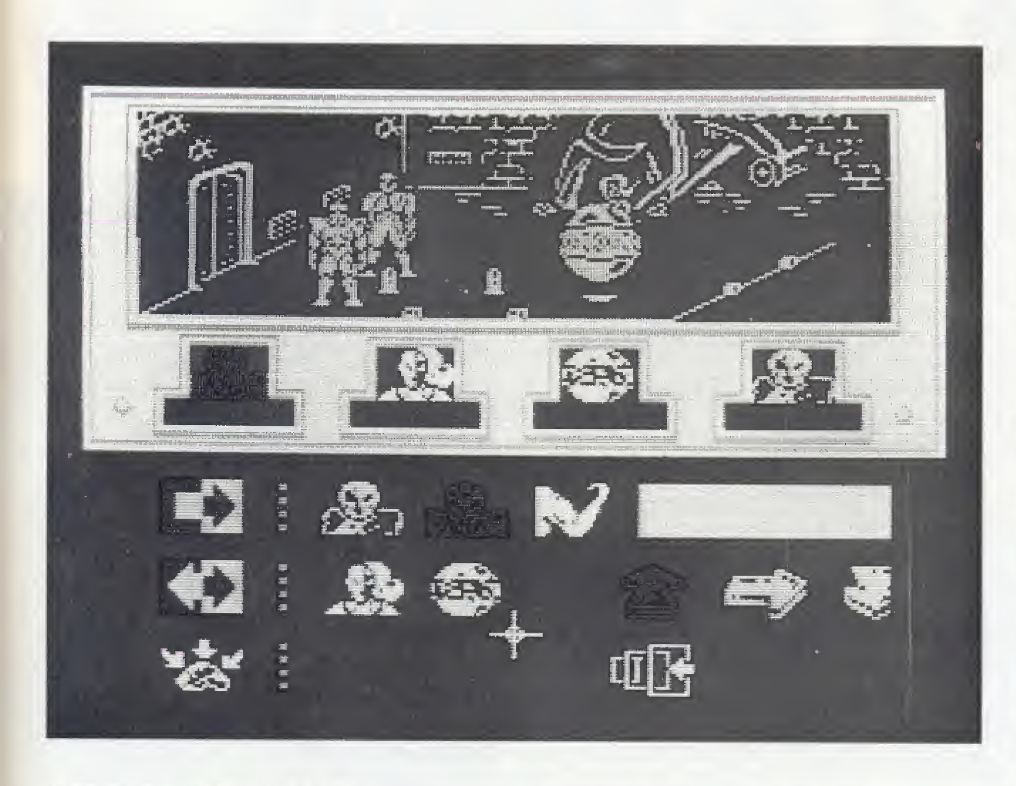
Pero he aquí que el simpático Phantomas consigue escapar de sus creadores y, siguiendo sus primarios instintos, se lanza a una carrera de ladrón profesional con el único fin de atesorar riquezas. Enterado de la existencia de un fabuloso tesoro que el avaro millonario Goldter ha acumulado en su mansión durante siglos, se dirige hacia allí dispuesto a librarle de tanto peso.

Nuestra misión es dirigir a este curioso ser a lo largo y ancho del planeta clónico Earth-Gamma, que se compone de tres partes principales: una base espacial con naves de distintas épocas que podremos usar para ciertas cosas, el palacio de Mr. Goldter, repleto de trampas, enemigos y toda clase de peligros, y el Mundo Subterráneo, habitado por todos los familiares del millonario que se encuentran encerrados en criptas.

Los movimientos que podemos dar al protagonista son los típicos del conocido Manic Miner. Al contrario que en la mayoría de los juegos de este tipo, las caídas desde grandes alturas no afectan en absoluto a nuestro héroe.

El nivel gráfico es bueno, denotando una cierta clase. Cabe destacar la variedad de formas y conjuntos, así como la extensión del mapa. Aunque no alcanza la perfección técnica de los grandes best-sellers del momento, puede catalogarse como un buen juego.



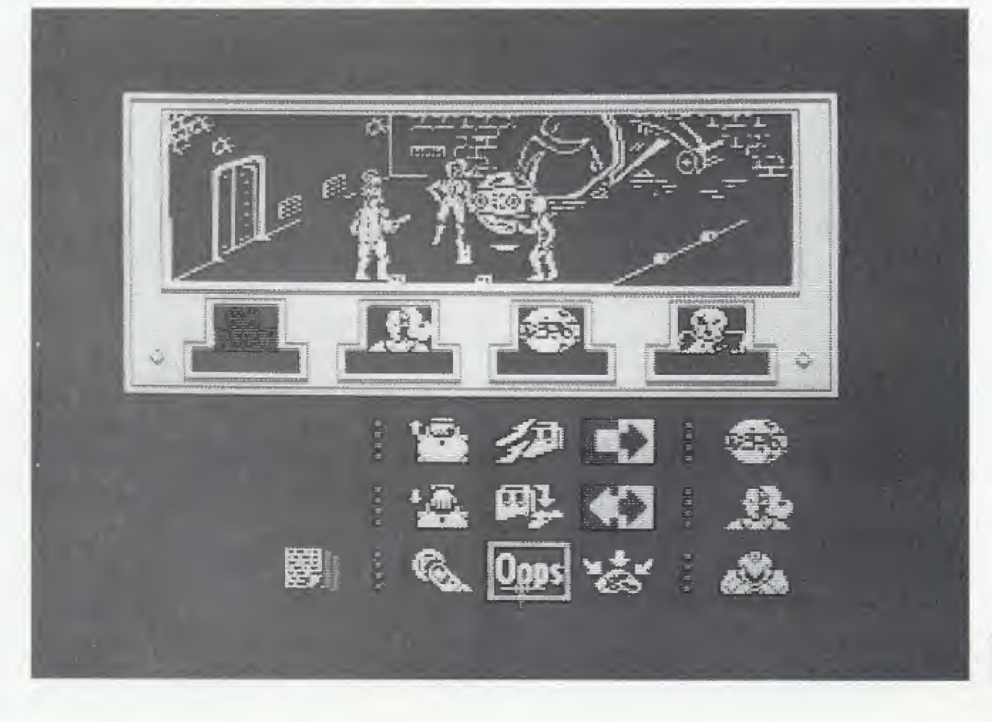


ENIGMA FORCE

BEYOND

SPECTRUM 48 K

Nos topamos en esta ocasión con uno de esos programas que merecen ser catalogados en un rincón especial. Continuación del conocido Shadowfire, una de esas «segundas partes» de la que se dice que nunca fueron buenas, Enigma Force va más allá de aquél, logrando unos niveles de calidad difícilmente superables.
Aunque el juego tiene más de aventura que de acción, no carece de esta, y consigue que se pasen las horas frente a él con una velocidad pasmosa. El hecho de que tengamos que manejar a cuatro personajes a un mismo tiempo influye bastante en que sea una aventura mucho más entretenida de lo habitual, y tenga ese interés especial que necesita este tipo de juegos para tener éxito.
Toda la estructura del programa es bastante innovadora. Lo más llamativo quizás sea la utilización que se hace de «íconos», seleccionables con un cursor dirigido con el joystick o el teclado, para manejar todas las opciones del juego. De esta forma el usuario no necesita aprender a qué tecla corresponde cada cosa, y puede guiarse por los dibujos y jugar sin preocuparse de las barreras ideomáticas en este sentido.
Desgraciadamente las instrucciones del juego no están traducidas, ni tampoco los escasos mensajes que aparecen a lo largo de la aventura.
Comienza a ser crónico que las casas de soft dejen de preocuparse de estos temas por buscar el negocio fácil.
En cuanto a todo lo demás, no puede achacársele gran cosa. Dispone de opciones para diferentes tipos de interfaces, joystick, buen sonido (incluida una larga, graciosa y polifónica melodía inicial), excelentes gráficos, velocidad, complejidad, un mapa completo, un argumento interesante,..., esperemos que sigan apareciendo juegos de este tipo de aquí en adelante.



CENTRONICS,

Escoger el interface Centronics ideal puede plantear mayores dudas que la elección de una impresora. Para ayudarte en tu decisión analizaremos tres de los más usados, los cuales representan tres formas de entender el concepto de interface.

Adquirir una impresora es un paso importante de cara a poseer un buen equipo informático. Poder obtener resultados en papel es muy interesante para ciertos trabajos, y a su vez, hay programas y aplicaciones en los que imprimir datos es objetivo prioritario (Tratamiento de textos, Hojas de cálculo, programas de diseño, etc.). De las característias de las impresoras, la velocidad de impresión es, y sobre todo de cara a su utilización en ordendores pequeños, el aspecto fundamental a la hora de decidir, incluso por encima de la calidad de letra. Para mejorar esta velocidad, se adoptó en los canales de comunicación ordenador-impresora la transmisión de los datos en *paralelo,* 8 bits a la vez, en vez de en *serie,* bit a bit. El SPECTRUM no posee una salida en paralelo, como otros ordenadores, para poder conectar periféricos, por lo tanto si vamos a utilizar una impresora que trabaje en paralelo necesitaremos un interface que nos permita comunicarnos con ésta.

Tres tipos de interfaces

Vamos a analizar tres interfaces *paralelo Centronics* para impresora.

El interface PIN necesita cargar software en memoria para funcionar correctamente

Cada uno de ellos tiene una forma particular de trabajar, posiblemente, cualquier otro interface de este tipo funcionará con el mismo esquema que alguno de estos tres. En primero lugar estudiaremos el CENTRONICS PIN, para poder trabajar con él hay que cargar en memoria desde cinta, y activar, una rutina en código máquina, que se suministra con el interface. El segundo es el CENTRONICS de INDESCOMP, en este caso el software necesario para controlar la impresora se encuentra en el propio interface, encargándose éste de colocarlo en memoria. Por último, el KEMPS-



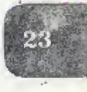
pero... ¿cual?

TON CENTRONICS realiza sus funciones desde el propio interface. Como vemos tres formas diferentes de realizar el mismo trabajo.

Se necesita un Spectrum 48K para utilizar el interface Centronics Indescomp

CENTRONICS PIN

Junto con este interface se suministra la cinta que contiene las rutinas que harán funcionar el interface. Sólo es necesario cargar una de éstas, cada una está adaptada para realizar copias de pantalla en una impresora de las más usuales en el mercado español. Estas rutinas sólo se diferencian en el uso de COPY, los comandos LPRINT y LLIST funcionarán en cualquier impresora, independientemente de la rutina cargada, de hecho, no hay rutinas para todas las impresoras, en aque-



llas impresoras para las que existe una rutina en la cinta (Cuadro 1). Con estas rutinas podemos ajustar el número de columnas a imprimir y

Se puede obtener un informe en pantalla del status del Kempston

cambiar el código de salto de línea (LF), mediante POKE's, pues estos controles están situados en el programa que controla el interface, que se encuentra en memoria. Para poder enviar códigos de control a la impresora (CR= retorno de carro, LF= salto de línea, ESC, etc.) es necesario asignar el código a un carácter gráfico, mediante POKE 64704+CODE "carácter gráfico",n; así estamos colocando el código de control cuyo número es n, en el gráfico que hay entre comillas. Ahora, basta imprimir el carácter gráfico que contiene el código de control para que la impresora realice la función de éste. Entre las rutinas que acompañan al interface hay una relocalizable. Su principal diferencia con las anteriores es que permite enviar códigos de control directamente con CHR$. Por ser relocalizable podemos situarla en cualquier parte de la memoria y así utilizar el interface con sólo 16K, pues las otras rutinas se ubican en los últimos octetos de las 32K superiores. También es posible asinar el código de LF, pero en cambio no puede realizar copias de pantalla, no podemos utilizar COPY. En el aspecto del diseño del interface hay que decir que no incorpora conector posterior, por lo que éste debe conectarse en último lugar en el port del SPECTRUM. Como punto negativo, hay que lamentar que el

Las copias de tamaño doble con el interface Kempston se hacen apaisadas

cable paralelo que conecta el interface con la impresora esté incrustado en el accesorio. Este es un gran fallo, pues si por cualquier motivo el cable se deteriora, necesitaremos cambiar todo el interface, pues es una parte solidaria de él.

Cuadro 2.

Informe en pantalla del estado del KEMPSTON CENTRONICS

```
SWITCHING COMMADS NO. 7F
COPY:REM
= 0TEXT
=* 1SEIK 100
=& 2SEIK 250
=# 3EPSON
=! 4EXTRA
CHR$ TOKENS 0–1
~ ENLARGED 0–1
27 (ESC) 0–1
/ LF STATUS 1 101
```

CENTRONICS INDESCOMP

Este interface tiene dos vías de comunicación, Paralelo Centronics y serie RS 232-C bidireccional. Ambas pueden conectarse a diferentes periféricos, una impresora y un MODEM. El software que lo hace funcionar está grabado en una *EPROM* (Erase and Programmable ROM = ROM programable), y al dar corriente al SPECTRUM, este soft es volcado en la parte superior de la memoria a partir de la dirección 64516. Por lo tanto sólo podemos utilizar este interface con un SPECTRUM de 48K. En su parte superior tiene un interruptor con dos posiciones, si se encuentra en la 2 al encender el ordenador, el software cargado en memoria es válido para impresoras como NEC, EPSON, C-ITOH, etc. Si está en la posición 1 el software sólo es válido para la GP-100 de SEIKOSHA. Una vez en memoria el software adecuado, debemos activarlo ejecutando RANDOMIZE USR 64973. Si fueramos a realizar copias de pantalla en impresoras EPSON, ADMATE, LOGITEC, deberíamos, además, colocar un 4 en la dirección 64254 (POKE

Cuadro 1. Relación de impresoras y tipos de COPY que proporcionan las rutinas del PIN I/F CENTRONICS.

IMPRESORAS	TIPO DE COPY
SEIKOSHA GP-550	COPY COPY de grises
SEIKOSHA SP-1000 SEIKOSHA SP-800 RITEMAN F+	COPY TAMAÑO NORMAL COPY TAMAÑO DOBLE
COMPATIBLES EPSON	COPY TAMAÑO NORMAL COPY TAMAÑO DOBLE

64254,4), para indicar el tipo de impresora gráfica que estamos utilizando.

A partir de este momento los comandos LPRINT y LLIST funcionarán correctamente para cualquier impresora, salvo que LPRINT no reconoce AT. Es posible seleccionar el envío de LF (salto de línea), pero para ello es necesario alterar parte del software contenido en la parte alta de la memoria mediante POKE's en 5 direcciones.

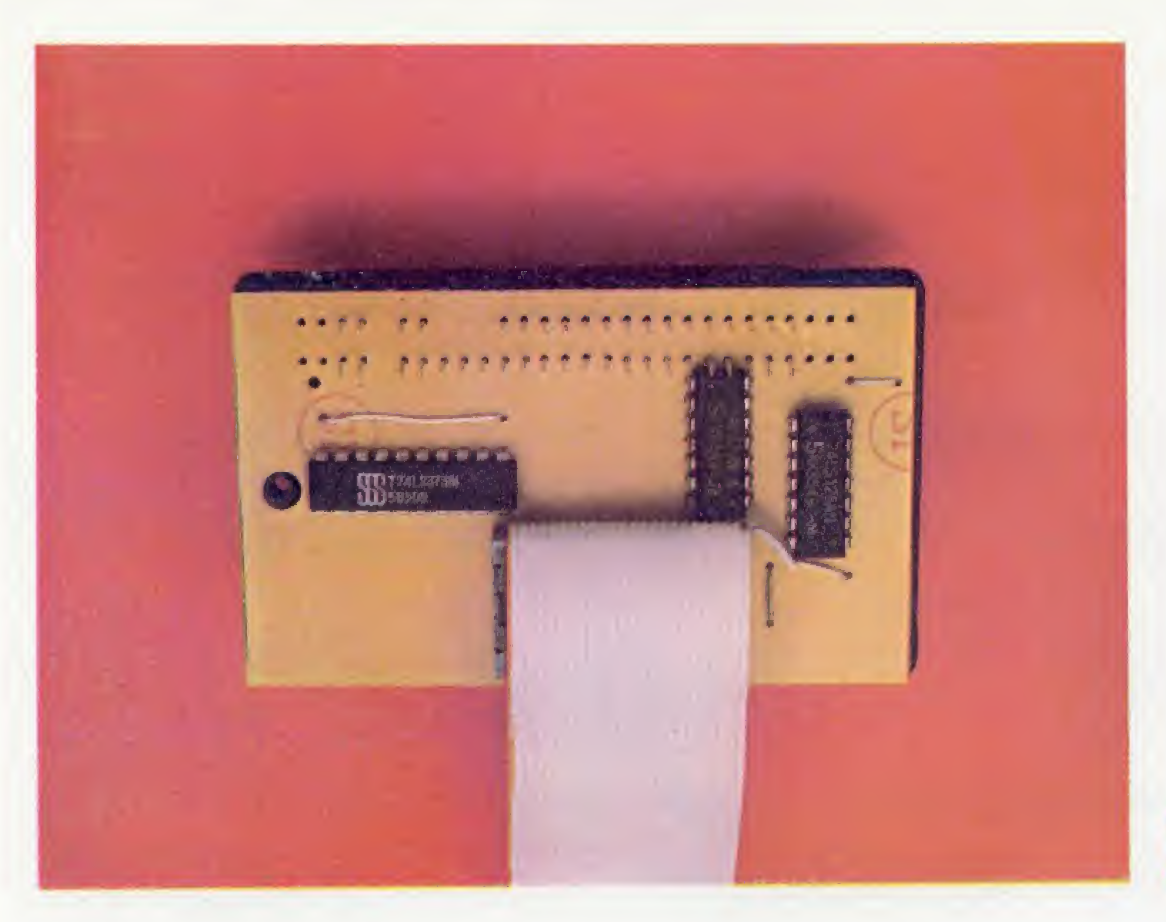
Para realizar copias de pantalla no se utiliza la instrucción COPY, para obtener una copia en tamaño normal ejecutaremos RANDOMIZE USR 65044, y en tamaño doble: RANDOMIZE USR 65047. En este tamaño las copias se imprimen con 32x16=512 puntos de anchura. Como las impresoras suelen tener para cada carácter una matriz de 7*5 puntos y 80 carácteres por línea, sólo disponen, por lo tanto, de 80*5=400 puntos de anchura, con lo cual en las copias tamaño grande faltan 112 puntos. el CENTRONICS INDESCOMP es el único de los tres que dispone de un conector posterior, por lo tanto puede colocarse en primer lugar en el port de expansión.

Lo más importante para el comprador es que el interface aproveche las funciones de su impresora

KEMPSTON CENTRONICS

Este interface presenta una concepción bastante diferente a los



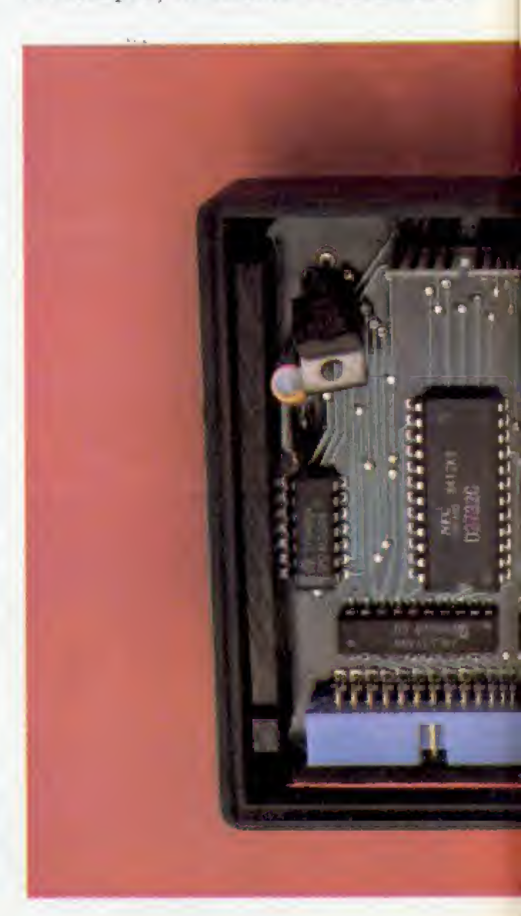


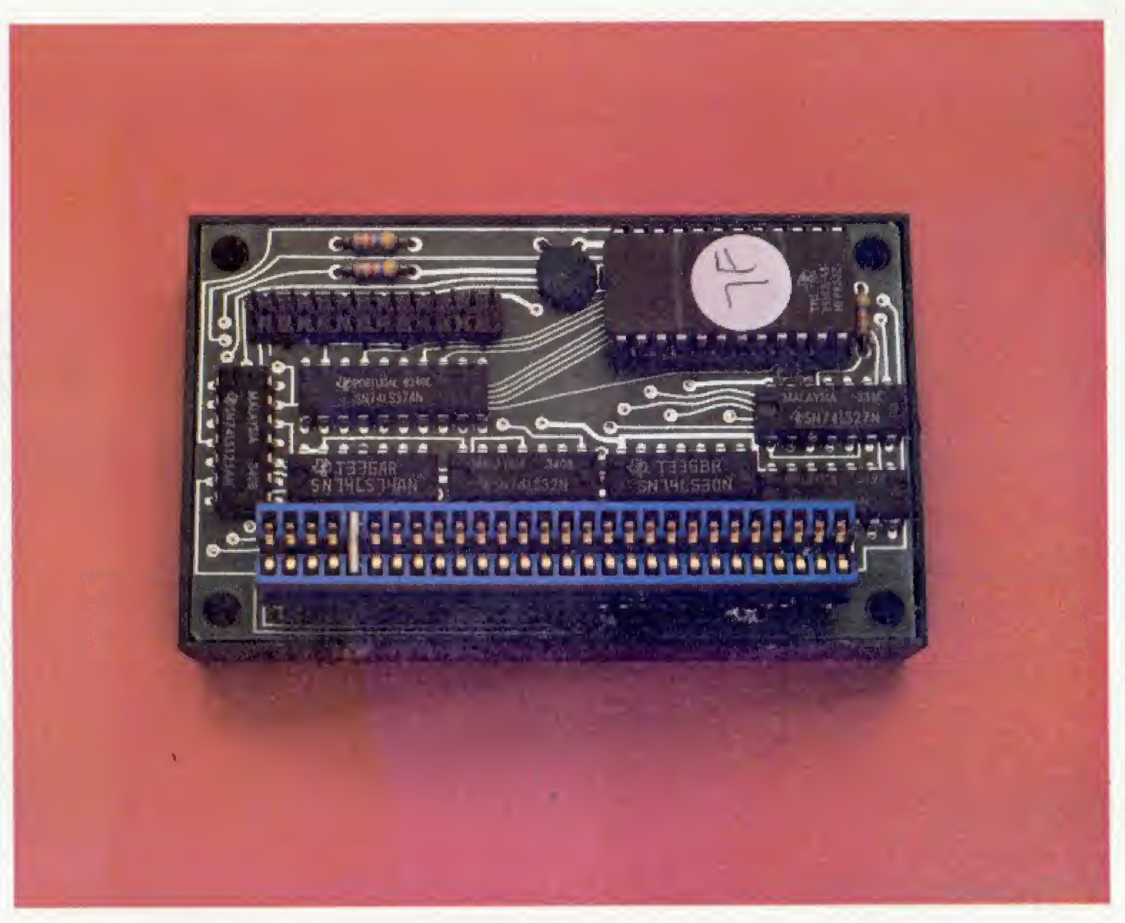
Interface Centronics de Pin Soft

dos anteriores. En aquellos era necesario cargar soft en memoria para controlar sus funciones, bien sea desde una cinta, o bien desde una EPROM del interface. En el KEMPSTON, por el contrario, es el propio interface el que se controla, por lo tanto la memoria queda libre. En aquellos, como los controles están en posiciones de memoria, podíamos cambiar ciertas características de impresión, con sólo hacer POKE en esas posiciones. Aquí, aunque esos controles están en el interface, también es posible. Tecleando COPY:REM?, obtenemos un informe en pantalla del status del interface (Cuadro 2). Primero, podemos seleccionar modo texto, o un tipo de impresora en particular. Si sólo vamos a utilizar los comandos LPRINT y LLIST podemos mantenernos en el modo 0 TEXT que hará trabajar copias de pantalla, debemos seleccionar en el interface la impresora con la que trabajamos, en este caso LPRINT, LLIST y COPY sólo funcionarán con esa impresora y sus similares. Estas opciones se activan con la instrucción COPY:REM = simb, donde simb representa a un tipo de impresora (espacio-TEXTO, *-SEIKOSHA 100, &-SEIKOSHA 250, =-EPSON y ! alguna otra impresora que no sea del tipo «SWITCH», interruptor, es deicr, tenemos dos opciones y sólo una de ellas puede estar activada.

Primero, CHR$-TOKENS, en el modo 0 está activa la opción CHR$, en este caso se imprimen los carácteres que representa cada código ASCII. De esta forma no podríamos listar un programa, pues es almacenado en memoria como TOKENS, y no con sus carácteres. Si ponemos este switch a 1 se activa la función TOKENS, de esta forma los TOKENS se amplían hasta obtener el comando al que representan, y así poder obtener listados. Esta función es la activa al encender el interface. La siguiente opción es la de copias en tamaño doble (ENLARGED), con la peculiaridad de que las copias grandes se realizan a lo largo, para no tener que truncar las últimas columnas. Es decir, los laterales de la pantalla son los que se copian en dirección horizontal, es una copia apaisada. Inicialmente está activa la copia normal. Es posible colocar o no el carácter de ESCAPE como 27, inicialmente es así. Y la última opción es la de LF, según sea necesario en nuestra impresora podemos hacer que el interface envíe el salto de línea (LF) o no.

Este interface no necesita inicialización, puesto que no carga software en memoria, por lo tanto está activo siempre, en cuanto es conectado.

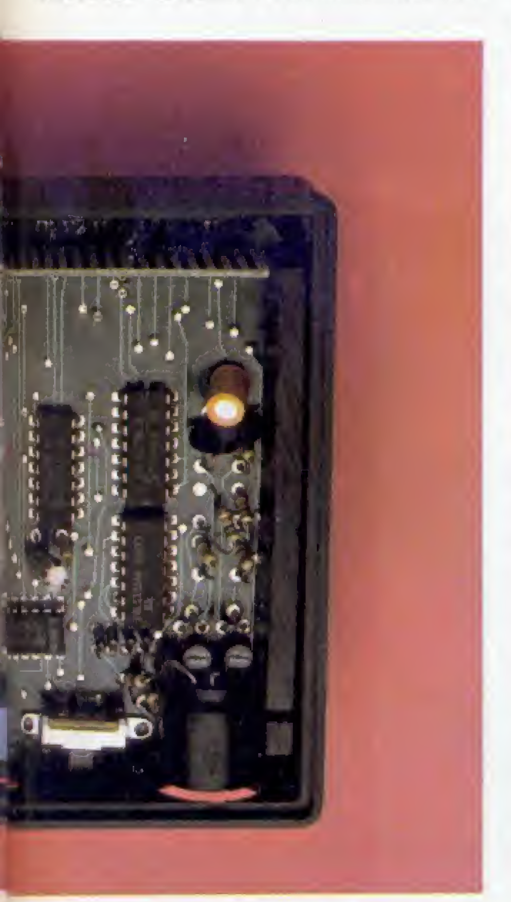




Interface Centronics Kempston E.

Crítica

Pasaremos revista ahora a los puntos positivos y negativos de cada interface. Del PIN resaltaremos las facilidades de COPY para un buen número de impresoras, así como la posibilidad de ajustar el número de columnas, que ninguno de los otros tiene. También que podemos utilizarlo con un SPECTRUM 16K, gracias a su rutina relocalizable. La otra cara de la moneda es que por posibilitar opciones de *hardcopy* a muchas impresoras, resulta bastante tedioso cargar el software desde la cinta cada vez que se utiliza el interface. El punto negro del diseño consiste en la incrustación del cable en la carcasa del interface.



Interface Centronics Indescomp.

El INDESCOMP no tiene grandes fallos, pero no tiene grandes virtudes, se mantiene en una gris línea media, realiza su trabajo sin alegrías. Es un punto a favor que aporte una salida RS 232-C, aunque casi nadie la pueda utilizar, y que tenga conector posterior. Su software ocupa memoria, pero no es necesario cargarlo desde cinta. El KEMPSTON es quizás el más perfeccionado, no aporta tantas variedades como el PIN, pero es fácil y muy cómodo de manejar. Gracias a su informe de estado en pantalla, sabemos siempre qué opciones tenemos activas, a la vez que realiza las funciones de menú de ayuda para los comandos de selección. No ocupa memoria y proporciona copias en tamaño grande apaisadas, sin cortar las últimas columnas, éstas son sus mejores cualidades, que quizás le colocan sobre los otros. En la parte negativa, sus pocas opciones de impresoras, aunque hay que tener en cuenta que fue uno de los interfaces CENTRONICS pioneros, el más antiguo de los tres que hoy tenemos.

Conclusión

Como hemos visto, son tres conceptos diferentes de hacer funcionar un interface. Los tres proporcionan aproximadamente las mismas posibilidades, quizás su diferencia más representativa es la facilidad que se otorga a una u otra impresora. Por esto, el aspecto más importante para el posible comprador, es que su impresora esté bien «atendida» por el interface. Si uno de estos proporcioona más opciones y aprovecha mejor las posibilidades de la impresora, esa debe ser su elección.

Joaquín Mateos Lago

TABLAS DE VERDAD

A quien en sus clases de filosofía se le haya atragantado alguna vez el cálculo lógico, es posible que le resulte útil este programa. Permite hallar la tabla de verdad de cualquier expresión lógica con hasta cinco variables proposicionales posibles.

El programa es corto y fácil de manejar; sólo hay que poner cuidado a la hora de copiarlo para hacerlo al pie de la letra y respetando cada número de línea. No sólo resultará útil para los interesados en la lógica proposicional y el álgebra de Bool, sino que también podría ser utilizado para comenzar con el estudio de la electrónica digital desde su base, las puertas lógicas.

Cuando lo ejecutemos, el programa nos pedirá unos segundos mientras copia el juego de carácteres en RAM, para poder así modificar algunos de ellos y conseguir los signos específicos que se utilizan para los operadores o conectores. Tras esto, aparecen en pantalla las teclas que habremos de usar para conseguir cada uno de estos signos y se nos pide que introduzcamos la expresión de la que queramos hayar su tabla. Esta expresión podrá utilizar hasta cinco variables, y cada una de éstas podrá repetirse varias veces; podrán usarse también todos los paréntesis que sean necesarios, siempre que se respete el orden correcto.

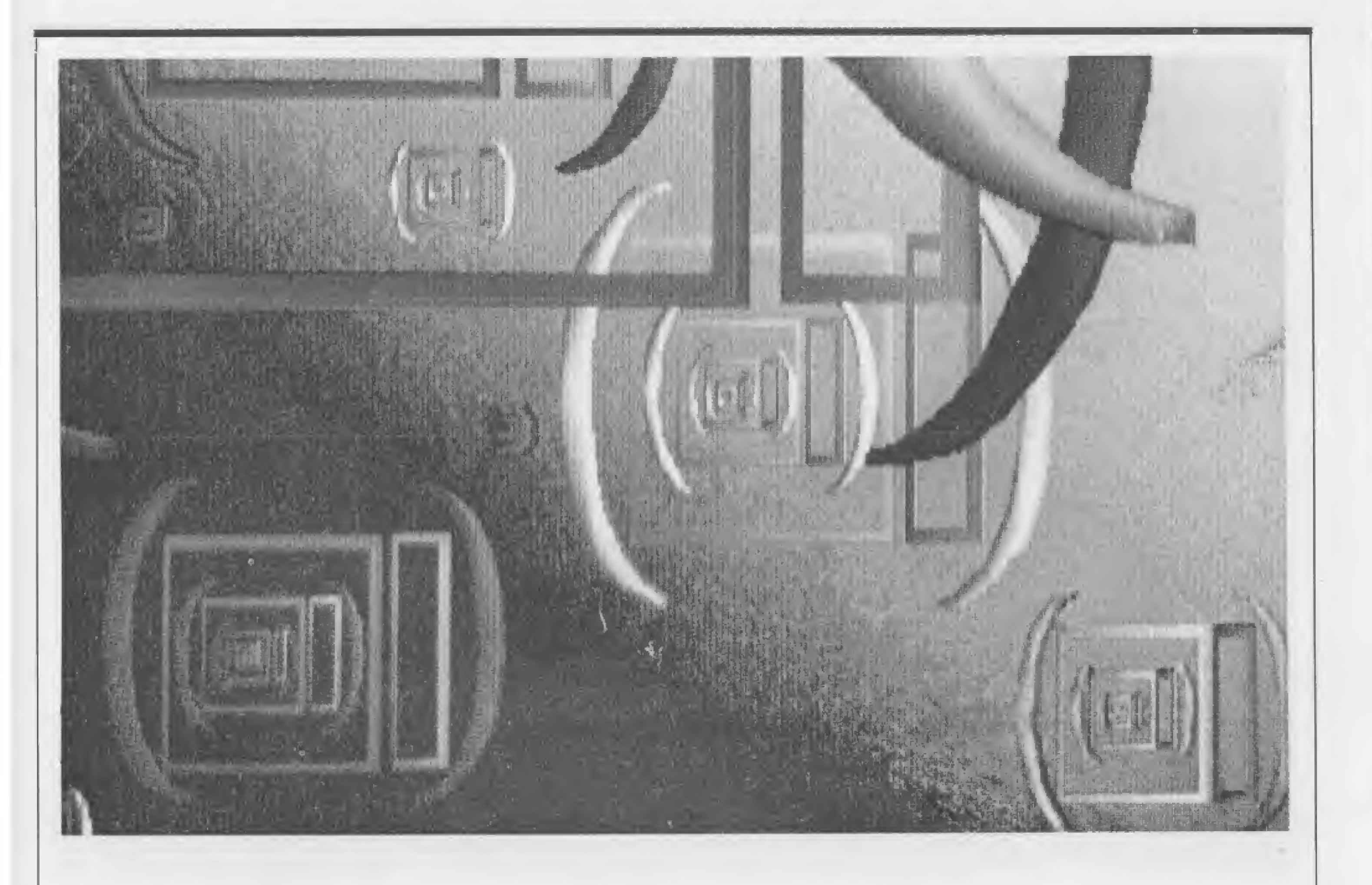
Cuando terminemos de teclear la expresión lógica, habrá que pulsar ENTER para que el ordenador se dé por aludido; tras esto el programa efectúa un chequeo de la expresión, indicando el error (si lo hubiera) con un interrogante tras el punto donde falle la sintaxis. Se tomará como no válido el utilizar variables distintas de p, q, r, t o s, usar como conector un carácter que no lo es, alterar el orden correcto de los elementos, no cerrar tantos paréntesis como se hayan abierto, etcétera. Si la expresión es correcta se da salida por pantalla a su correspondiente tabla de verdad y se espera a que se pulse una tecla para comenzar de nuevo pidiendo una nueva expresión.

Sólo resta desearos que podáis convencer a vuestro profesor para que os permita llevar el Spectrum a vuestros exámenes de Filosofía.

Luis Gala

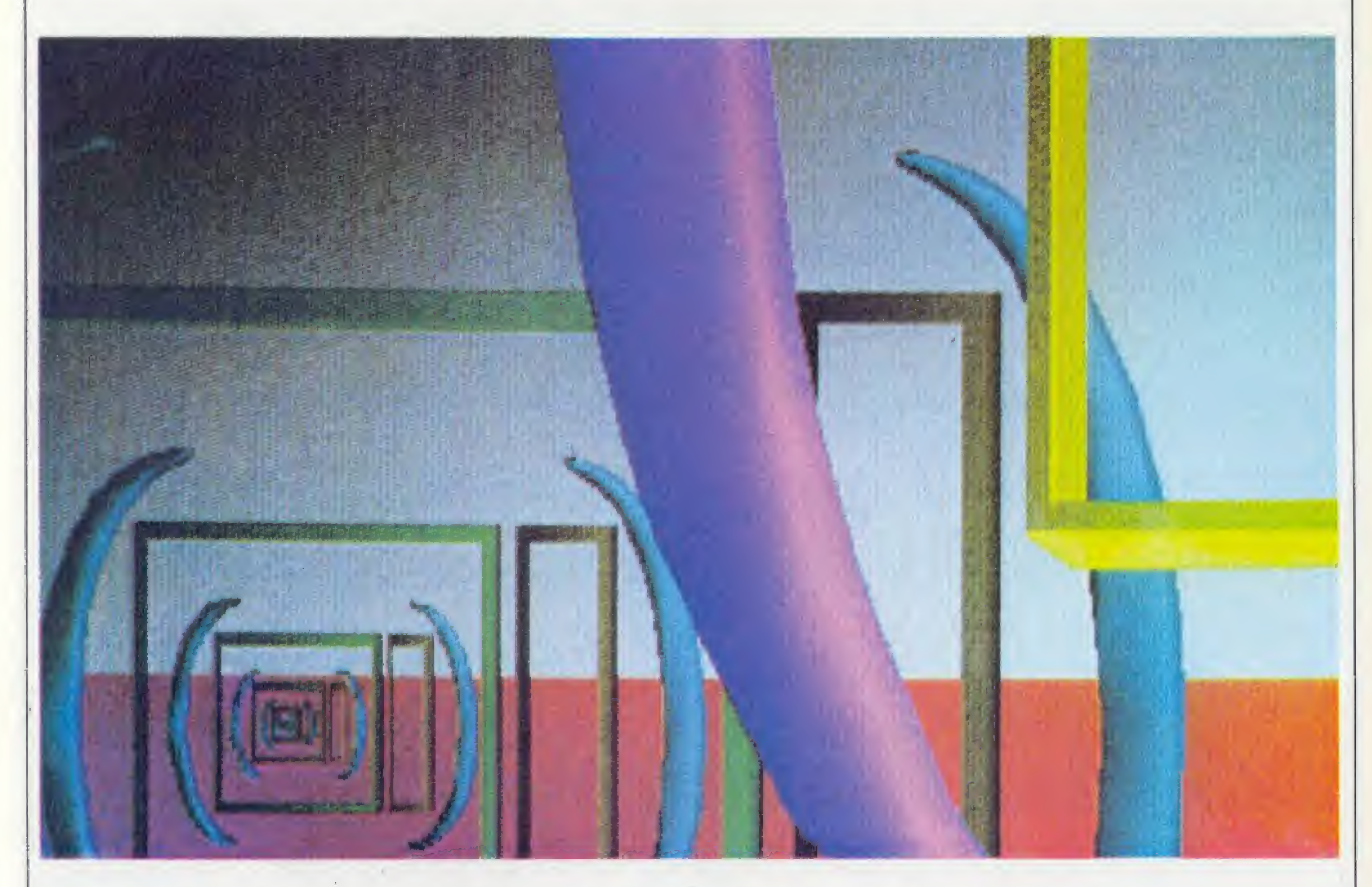
```
 10 REM  TABLAS DE VERDAD

 20 POKE 23658,0: OVER 1
 30 PRINT  FLASH 1;AT 10,4;" UN
MOMENTO, POR FAVOR "
 35
 40 REM modificar juego caract.
 45
 50 FOR n=0 TO 770
 60 POKE 30000+n,PEEK (15616+n)
 70 NEXT n
 80 FOR n=0 TO 2: READ x
 90 FOR m=0 TO 5: READ y
100 POKE 30000+x+m,y
```



```
 110 NEXT m: NEXT n
 115
 120 REM instrucciones
 125
 130 CLS : BEEP .1,30
 135 PRINT '"           TABLAS DE V
ERDAD"'"          PATATITA SOFT"
 140 PRINT '" INTRODUCE UNA EXPR
ESION LOGICA    UTILIZANDO ESTAS
 TECLAS:"
 150 PRINT '"    ""y"" como conju
ntor"
 160 PRINT '"    ""v"" como disyu
ntor inclusivo"
 170 PRINT '"    ""w"" como disyu
ntor exclusivo"
 180 PRINT '"    ""-"" como negad
or"
 190 PRINT '"    ""^"" como condi
cional"
 200 PRINT '"    ""="" como bicon
dicional"
 210 PRINT '"  Puedes utilizar p
arentesis y      p,q,r,s,t como
variables."
 220 POKE 23606,48 :  POKE 23607
,116
```

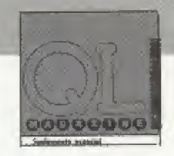
```
 225
 230 REM input-test
 235
 240 LET vars=0: LET c$=""
 250 LET y=1: LET z=0
 260 INPUT  LINE a$
 270 IF a$="" THEN  GO TO 260
 280 LET b$=a$
 290 FOR n=1 TO LEN a$
 300 LET x=CODE a$(n)
 310 IF x<33 OR x=40 OR x=41 THE
N  LET z=z-(x=40)+(x=41) : GO TO
 480-40*(NOT y AND x=40 OR y AND
 x=41)
 320 IF x=45 THEN  LET b$(n)=CHR
$ 195: GO TO 470-30*NOT y
 330 LET y=y+1
 340 IF x=61 THEN  GO TO 470
 350 IF x=118 THEN  LET b$(n)=CH
R$ 197: GO TO 470
 360 IF x=121 THEN  LET b$(n)=CH
R$ 198: GO TO 470
 370 IF x=119 THEN  LET b$(n)=CH
R$ 201: GO TO 470
 380 IF x=94 THEN  LET b$(n)=CHR
$ 199: GO TO 470
 390 FOR m=1 TO vars
```



```
 400 IF x=CODE c$(m) THEN  LET y
=y-2: GO TO 470
 410 NEXT m
 420 IF x>111 AND x<117 THEN  LE
T c$=c$+CHR$ x: LET vars=vars+1:
 LET y=y-2: GO TO 470
 425
 430 REM error sintaxis
 435
 440 PRINT )0;a$( TO n); FLASH 1
;"?"; FLASH 0;a$(n+1 TO )
 450 BEEP .2,-10: PAUSE 0
 460 GO TO 240
 470 IF y=-1 OR y=2 OR z=1 THEN
 GO TO 440
 480 NEXT n
 490 IF y OR z THEN  LET n=n-1:
GO TO 440
 495
 500 REM ejecucion
 505
 510 CLS : PRINT ;"  ";a$;''''"
";c$''
 520 FOR n=0 TO 2^vars-1
 530 IF n=16 THEN  PRINT AT 3,18
;c$'',
 540 LET y=n
 550 LET d$=""
 560 FOR m=1 TO vars
 570 LET x=INT (y/2)
 580 LET d$=STR$ (x<>y/2)+d$
 590 LET y=x
 600 NEXT m
 610 FOR m=1 TO vars
 620 LET x=VAL d$(m)
 630 IF c$(m)="p" THEN  LET p=x
 640 IF c$(m)="q" THEN  LET q=x
 650 IF c$(m)="r" THEN  LET r=x
 660 IF c$(m)="s" THEN  LET s=x
 670 IF c$(m)="t" THEN  LET t=x
 680 NEXT m
 690 PRINT "  ";d$;"    ";VAL b$,
,
 700 NEXT n
 710 BEEP .1,30: PRINT )0;TAB 6;
"PULSA UNA TECLA"
 720 POKE 23606,0: POKE 23607,60
 730 PAUSE 0: GO TO 130
 735
 740 REM caracteres modificados
 745
 750 DATA 233,0,36,66,255,66,36
 760 DATA 714,16,40,40,68,68,0
 770 DATA 497,0,8,4,254,4,8
```



El conjunto de Mandelbrot



El conjunto de Mandelbrot

LISTADO 1. Fractal copo de nieve

```
100 WINDOW 512,256,0,0
110 MODE 4:PAPER 0
120 CLS
130 PENDOWN:TURNTO 0
140 LINE 40,80
150 FOR j = 0 TO 3
160   FOR i=1 TO 4
170     linea j,60
180     TURN -90
190   END FOR i
200   PAUSE
210 END FOR j
220 STOP
230 :
240 DEFine PROCedure linea (n,long)
250   LOCal long4
260   IF n = 0
270     MOVE long
280     RETurn
290   ELSE
300     long4 = long / 4
310     linea n-1,long4
320     TURN 90
330     linea n-1,long4
340     TURN -90
350     linea n-1,long4
360     TURN -90
370     linea n-1,long4
380     linea n-1,long4
390     TURN 90
400     linea n-1,long4
410     TURN 90
420     linea n-1,long4
430     TURN -90
440     linea n-1,long4
450   END IF
460 END DEFine linea
```

¿Cuánto hay de aquí al puerto más cercano? —preguntó el pescador desde su barca al caminante. Depende cuán próximo vaya a la orilla —contestó éste.

Esta extraña conversación entre un pescador y un caminante nos permite introducirnos en un tema fascinante y complejo —¿Cuál es la longitud de la costa de un país?— De la respuesta del caminante podemos deducir que depende de cuán finamente midamos. Es un hecho cierto, que al observar un mapa de una costa a gran escala vemos una distribución de cabos y golfos, asombrósamente parecida a la de un mapa visto a pequeña escala; en los dos casos aparecen entrantes y salientes de la misma escala de magnitud que la de observación. Se dica pues que la costa marítima es una línea fractal, esto es, que independientemente de cuán cerca la observemos presenta siempre el mismo aspecto.

No es único este ejemplo. En matemáticas se conocen desde hace mucho tiempo definiciones para las curvas que integran objetos más o menos extraños como son curvas a puntitos, o con picos.

Cada vez que se encontraban los matemáticos con objetos que parecían curvas, pero que no eran «curvas a simple vista» era preciso ampliar la definición del concepto de curva para incluir o eliminar estos nuevos objetos.

Cuando aparecieron curvas con picos, los matemáticos pensaron que estas curvas se portarían razonablemente bien, ¡cuán equivocados estaban! En la segunda mitad del siglo pasado empezaron a aparecer curvas que tenían picos ¡¡en todos sus puntos!! Una de las curvas más extrañas fue la presentada en 1890 por **Giuseppe Peano.** Esta curva tenía la propiedad de pasar por todos los puntos interiores y de la frontera de un cuadrado que la contenía, y si bien la curva es totalmente continua (no contiene saltos) es totalmente imposible determinar la dirección que seguirá en ninguno de sus puntos.

Posteriormente aparecieron otras curvas de este tipo, curvas a las que se dio el nombre genérico de «Curvas de Peano», de las que es corriente encontrarse ejemplos de curvas abiertas y cerradas.

En el caso de las curvas de Peano cerradas (como las gomas elásticas), se presenta otra dificultad, no se puede distinguir qué puntos del cuadrado están dentro y cuáles fuera de la superficie encerrada por la curva.

Para ver un ejemplo de una de estas fronteras exóticas sígase la siguiente receta:

Cójase un cuadrado de tamaño medio. Métase en la cacerola a fuego... (...) Un momento, creo que me estoy liando.

Cójase un cuadrado de tamaño medio. Tómese cada una de sus cuatro aristas y (sí, ahora va bien) divídase en cuatro trozos iguales. Sustituyanse los dos trozos interme-



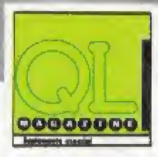
dios por sendos cuadrados de arista la longitud de uno de los trozos que estamos eliminando (y eliminando el lado que reposa sobre la arista inicial), dejando uno de los cuadrados hacia adentro y otro hacia afuera. Obtenemos así una curva en «copo de nieve cuadrado», que es generada por uno de los programas que acompañan este artículo (para ver la relación, dependiendo de la profundidad no se borran las fases iniciales, cosa que puede remediarse con un buen CLS).

Iterando obtenemos una curva muy curiosa con las propiedades siguientes: el perímetro del cuadrado es ahora de longitud infinita (tan grande como se quiera) pues en cada iteración hemos doblado su longitud; sin embargo, aquí está lo sorprendente: la superficie encerrada por la nueva frontera es exáctamente igual a la del cuadrado original por haber añadido en cada iteración lo mismo que se quita.

Lo mismo que en el caso de curvas, el comportamiento fractal aparece en muchos otros objetos matemáticos, entre otros el llamado con-

Listado 2. Conjunto de Mandelbrot versión lenta

```
100 inicializa
110 REPeat lazo
120   rellena
130   aumenta
140 END REPeat lazo
150 :
160 DEFine PROCedure rellena
170   dx=dxi/256
180   dy=dyi/256
190   FOR i=0 TO 255
200     xx = XI + dx * i
210     yy1 = yi + 255 * dy
220     yy2 = yi + 128 * dy
230     bisecy INT (i),0,127,HUIDA (xx,yy1),HUIDA (xx,yy2)
240     yy1 = yi + 127 * dy
250     yy2 = yi
260     bisecy INT (i),128,255,HUIDA (xx,yy1),HUIDA (xx,yy2)
270   END FOR i
280 END DEFine rellena
290 :
300 DEFine PROCedure bisecy (x%,y1%,y2%,c1%,c2%)
310   LOCal xcc,yc1,yc2
320   xcc = XI + dx * x%
330   yc1 = yi + dy * (255 - INT((y1%+y2%)/2))
340   yc2 = yi + dy * (255 - INT((y1%+y2%)/2)-1)
350   IF c1% = c2% THEN
360     linea x%,y1%,x%,y2%,c1%
370   ELSE
380     IF (y2%-y1%) = 1 THEN
390       BLOCK 2,1,2*x%,y1%,color% (c1%)
400       BLOCK 2,1,2*x%,y2%,color% (c2%)
410     ELSE
420       bisecy x%,y1%,INT((y1%+y2%)/2),c1%,HUIDA(xcc,yc1)
430       bisecy x%,INT((y1%+y2%)/2)+1,y2%,HUIDA(xcc,yc2),c2%
440     END IF
450   END IF
460 END DEFine bisecy
470 :
480 DEFine PROCedure linea (xa%,ya%,xb%,yb%,c%)
490   BLOCK (2*(xb%-xa%+1)),(yb%-ya%+1),(2*xa%),ya%,color% (c%)
500 END DEFine linea
510 :
520 DEFine PROCedure inicializa
530   MODE 8:WINDOW 512,256,0,0
540   PAPER 2:CLS:CSIZE 3,1
550   INPUT '   #x?''XI\'   #y?''yi\'  #dx?''dxi\'  #dy?''dyi
560   PAPER 0:CLS
570 END DEFine inicializa
580 :
590 DEFine FuNction color% (x%)
600   IF x%=1000 THEN RETurn (0)
610   IF x% > 300 THEN RETurn (7)
620   IF x% > 100 THEN RETurn (6)
630   IF x% > 32 THEN RETurn (4)
640   RETurn (1)
650 END DEFine color%
660 :
670 DEFine PROCedure aumenta
680   x% = 0
690   y% = 0
700     box
710   END REPeat loza
720 END DEFine aumenta
730 :
740 DEFine PROCedure box
750   BLOCK dx%-1,1,x%,y%,7
760   BLOCK 2,dy%,x%+dx%-2,y%,7
770   BLOCK dx%-1,1,x%,y%+dy%-1,7
780   BLOCK 2,dy%,x%,y%,7
790 END DEFine box
800 :
810 DEFine PROCedure nuevo
820   ddx = dxi / 512
830   ddy = dyi / 256
840   XI = XI + x% * ddx
850   yi = yi + (256-y%-dy%+1)*ddy
860   dxi = dxi * dx% / 512
870   dyi = dyi * dy% / 256
880 END DEFine nuevo
```





junto de Mandelbrot, llamado así en honor de **Benoit B. Mandelbrot,** investigador de IBM en Nueva York, que ha impulsado enormemente el estudio de las figuras fractales como la mostrada antes. En el caso del conjunto de Mandelbrot, calificado por **John H. Hubbard** de la Universidad de Cornell como «El objeto más complicado de las matemáticas», no es una curva la que muestra comportamiento fractal, o por lo menos no una curva en el sentido tradicional, pues se trata de un conjunto de punto del plano complejo, y si bien su representación es un poco sosa, haremos una serie de pantallas bonitas que se podrá bien fotografiar o bien sacar a impresora con una rutina como la que se publicó en un número anterior de TODOSPECTRUM.

Como el conjunto de Mandelbrot aparece trabajando con números complejos, es conveniente repasarlos un poco, para ello hemos incluido un apéndice al final del artículo con algunos de los conceptos y fórmulas más utilizados en lo que viene a continuación.

Sea la función que toma un número complejo y nos devuelve otra dada por $f(z)=z^2+c$ donde c es una constante y z un número complejo. Si partimos inicialmente de z=0+0i y el resultado de la función lo volvemos a introducir en ella, obtendremos una sucesión de números complejos dada por:

$$c;\ c^2+c;\ (C^2+c)^2+c;...$$

Dependiendo del valor que inicialmente demos a la constante c, obtendremos una sucesión u otra. La pregunta del millón de créditos intergalácticos es: ¿Para qué valores de la constante c, la sucesión está acotada?, dicho de otra forma, se trata de encontrar los valores de c que hagan que la sucesión que genera se quede siempre razonablemente cerca del punto 0+0i. Estos puntos son los que van a pertenecer al conjunto de Mandelbrot.

Para medir cuanto de cerca está un número complejo del origen, se toma lo que se conoce como su módulo, que se corresponde con la raíz cuadrada de la suma del cuadrado de la parte real más el cuadrado de la parte imaginaria.

Por unos resultados de la teoría de funciones complejas se demuestra que si un elemento de la sucesión tiene un módulo mayor que 2, entonces a la larga los módulos de los números de la sucesión crecerán de manera ilimitada, con lo que podemos afirmar que los valores de c que en un elemento de la sucesión por ellos creados excede su módulo del valor 2 no pertenecen al conjunto que estamos intentando delimitar.

Sin embargo sólo hemos dicho que algunos valores de c no pertenecen al conjunto. Es difícil decir a priori si un valor dado de c pertenece o no al conjunto que estamos buscando pues nada nos dice si después de 500 iteraciones el módulo aún es menor que 2 que la iteración 501 no se nos vaya. Por esto se suele dar un límite máximo de iteracio-

Listado 3. Programa fuente en ensamblador

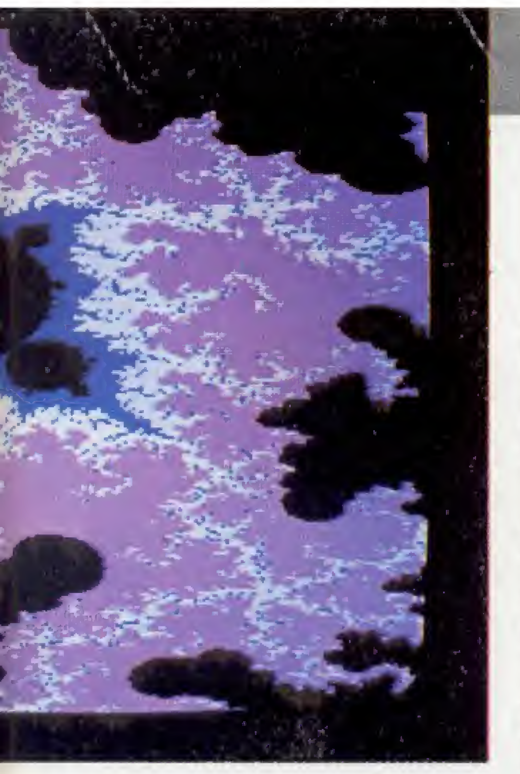
```
;-----------------------------------------------------------------
;       Esta rutina assembler, lista para ser llamada por el BASIC
; permite ampliar zonas del conjunto de Mandelbrot.
;
;       Para ello, se teclea MUIDA x,y y devuelve el numero que
; hace falta para que su modulo pase de 2.
;
;       Ver Investigacion y Ciencia de Octubre de 1.985
;
;               Copyright Gerardo Izquierdo 10-85
;-----------------------------------------------------------------

;-----------------------------------------------------------------
;       Igualdades iniciales
;-----------------------------------------------------------------

;---------------- Vectores ---------------------------------------
ut_scr          equ     $C8
ut_err0         equ     $CA
bp_init         equ     $110
ca_gtfp         equ     $114
bv_chrix        equ     $11A
ri_exec         equ     $11C
ri_execb        equ     $11E
;---------------- Trap #2 ----------------------------------------
io_open         equ     $1
io_close        equ     $2
;---------------- Trap #3 ----------------------------------------
sd_fill         equ     $2E
;---------------- Errores ----------------------------------------
err_bp          equ     -15
;---------------- SuperBasic -------------------------------------
bv_rip          equ     $58
;-----------------------------------------------------------------
;       Inicializar el procedimiento
;-----------------------------------------------------------------
inicio  lea.l   proc_def,a1
        move    bp_init,a2
        jsr     (a2)
        rts

;-----------------------------------------------------------------
;   Tabla de definicion de procedimientos
;-----------------------------------------------------------------
proc_def dc.w   0                ; 0 procedimientos
        dc.w    0
        dc.w    1                ; 1 funcion
        dc.w    mandel-*      ; comienzo
        dc.b    5,'MUIDA'        ; nombre
        align
        dc.w    0,0

;-----------------------------------------------------------------
;       Cargar los valores de x,y
;-----------------------------------------------------------------
mandel
        move.l  #60,d1          ; Reservamos sitio para 10 numeros
        move.w  bv_chrix,a2
        jsr     (a2)
        move.w  ca_gtfp,a2       ; poner en el stack los parametros
        jsr     (a2)
        dc.b    l_x              ; x x1*x1-y1*y1 x1
        dc.b    $A               ; x1*x1-y1*y1+x x1
        dc.b    s_x1             ; x1
        dc.b    l_y1             ; y1 x1
        dc.b    $E               ; x1*y1
        dc.b    $16              ; x1*y1 x1*y1
        dc.b    $A               ; 2*x1*y1
        dc.b    l_y              ; y 2*x1*y1
        dc.b    $A               ; 2*x1*y1+y
        dc.b    $16              ; 2*x1*y1+y 2*x1*y1+y
        dc.b    s_y1             ; 2*x1*y1+y
; En estos momentos, x1 e y1 tienen el nuevo valor.
        dc.b    $16              ; y1 y1
        dc.b    $E               ; y1*y1
        dc.b    l_x1             ; x1 y1*y1
        dc.b    $16              ; x1 x1 y1*y1
```



nes, pasadas las cuales sin que el módulo del elemento en cuestión sobrepase 2, declaramos «por definición» que el punto representado por c pertenece al conjunto.

Para hacer una representación del conjunto en la pantalla del QL será necesario efectuar una correspondencia de los pixels de la pantalla con los puntos del plano complejo, y colorear estos pixels según un criterio, criterio que puede ser necesario modificar.

Damos a continuación dos versiones del programa: una de ellas es para largos períodos de tiempo y la otra es para gente con prisas.

En las dos versiones se nos pide que demos las coordenadas de la esquina inferior izquierda de la pantalla y la longitud y altura del rectángulo que va a ser la pantalla sobre el plano complejo.

Las dos versiones del programa utilizan una única subrutina en código máquina encargada de efectuar el grueso de los cálculos. A esta función se le pasan como parámetros las coordenadas del punto complejo c, y devuelve un número entero entre 1 y 1000 que representa el número de iteraciones que han sido necesarias para que el módulo del elemento en cálculo de la sucesión sobrepasase el valor de 2. Realmente, para evitarse una raíz cuadrada, compara el cuadrado del módulo con 4, que nos da lo mismo.

El código de la rutina está dado tanto en asembler como en un programa cargador en SuperBASIC para aquellos que no posean un asembler, siendo en este caso necesario el teclearse el programa basic poniendo mucho cuidado, y ejecutándolo por «RUN», lleva a cabo una prueba elemental del contenido de las líneas «DATA» y se graba el resultado en el fichero «HUIDA_CDE», que será necesario cargar con:

```
a=RESPR(256):LBYTES
mdv1_huida_cde,a:CALL a
```

Antes de ejecutar cualquiera de los programas BASIC que se explican a continuación. El motivo de que la carga no vaya en los programas es que no es infrecuente el estar dando «RUN» al programa para cambiar de imagen, y cogería cada vez memoria, por lo que es más conveniente de esta forma y se tarda sólo un momento.

Esta función hace uso abundante del paquete de instrucciones de coma flotante que suministra el SuperBASIC, intentando optimizar el uso del juego de instrucciones, al punto de hacer tan solo una llamada previa, y otra llamada a la rutina de cálculos, esta vez dentro de un bucle que tiene como límite el ya señalado de 4 ó de 1.000 iteraciones.

Se podrá advertir un pequeño error, caso de que el punto pasado como parámetro tenga un módulo mayor que 2, se devuelve 1 en vez de cero. Sinceramente no he corregido este problema por no afectar en absoluto a los resultados, efectivamente las zonas del plano complejo que se exploran devuelven valores altos de la función.

```
        dc.b    $E              ; x1*x1 y1*y1
        dc.b    $A              ; x1*x1+y1*y1 = R
        dc.b    l_4             ; 4 R
        dc.b    $C              ; R-4
        dc.b    0               ; fin
;---------------------------------------------------------------
;
;                  zona de variables
;             +----------------------------+ <-- (A6,A4.L)
;             !             y              !
;             +----------------------------+
;             !             x              !
;             +----------------------------+
;             !             y1             !
;             +----------------------------+
;             !             x1             !
;             +----------------------------+
;             !             4              !
;             +----------------------------+
;---------------------------------------------------------------
        cmp.w   #2,d3           ; ¿Hay dos parametros?
        bne     err_bp          ; NO! error
        clr.l   -4(a6,a1.l)     ; ponemos 2 ceros (x1 e y1).
        clr.l   -8(a6,a1.l)
        clr.l   -12(a6,a1.l)
        move.l  a1,a4
        add.l   #12,a4
        sub.l   #18,a1          ; actualizamos el puntero de pila
        move.l  #$40000000,2(a6,a1.l)     ; ponemos 4
        move.w  #$0803,0(a6,a1.l)
        clr.l   d4              ; empezamos a contar
lazo1
        addq.w  #1,d4           ; a    adimos 1 al contador
        moveq   #0,d7           ; siempre cero
        lea.l   calc_l,a3       ; lista de calculos
        move.w  ri_execb,a2     ; orden de calcular
        jsr     (a2)            ; R-4 en 0(A6,A1.l)
        tst.l   d0
        bne.s   retorno
        add.w   #6,a1           ; restauramos pila
        tst.l   -4(a6,a1.l)     ; miramos signo de R-4
        bge.s   final           ; si >=0 terminar
        cmp.w   #1000,d4        ; ¿1000 iteraciones?
        blt.s   lazo1           ; NO seguir
;---------------------------------------------------------------
;       Devolvemos el resultado y terminamos
;---------------------------------------------------------------
final
        moveq   #0,d0
        move.l  a4,a1           ; tomamos el indice bueno de la pila
        subq    #2,a1           ; a    adimos 2 octetos
        move.l  a1,bv_rip(a6)   ; salvamos el nuevo valor
        move.w  d4,0(a6,a1.l)   ; cargamos ahi el numero de iter.
        moveq   #3,d4           ; el tipo es entero
        rts                     ; terminamos
;---------------------------------------------------------------
;       Error y retorno
;---------------------------------------------------------------
err_bp
        moveq.l #err_bp,d0
retorno rts
;---------------------------------------------------------------
;       Definiciones de los calculos
;---------------------------------------------------------------
l_r             equ     -12
l_y             equ     -6
l_x1            equ     -24
l_y1            equ     -18
s_x1            equ     -23
s_y1            equ     -17
l_4             equ     -30
calc_l  dc.b    l_x1            ; x1
        dc.b    $16             ; x1 x1
        dc.b    $16             ; x1 x1 x1
        dc.b    $E              ; x1*x1 x1
        dc.b    l_y1            ; y1 x1*x1 x1
        dc.b    $16             ; y1 y1 x1*x1 x1
        dc.b    $E              ; y1*y1 x1*x1 x1
        dc.b    $C              ; x1*x1-y1*y1 x1
```



Listado 4. Conjunto de Mandelbrot versión rápida.

```
100 inicializa
110 REPeat lazo
120   rellena
130   aumenta
140 END REPeat lazo
150 :
160 DEFine PROCedure inicializa
170   factor = 8
180   MODE 8:WINDOW 512,256,0,0
190   PAPER 2:CLS:CSIZE 3,1
200   INPUT '   #x?''XI\'   #y?''yi\'  #dx?''dxi\'  #dy?''dyi
210   PAPER 0:CLS
220 END DEFine inicializa
230 :
240 DEFine PROCedure rellena
250   CLS
260   dx = dxi * factor / 256
270   dy = dyi * factor / 256
280   x0=XI-dx
290   y=yi-dy
300   FOR j=0 TO 255 STEP factor
310     y=y+dy
320     x=x0
330     FOR i=0 TO 511 STEP 2*factor
340       x=x+dx
350       BLOCK 2*factor,factor,i,256-factor-j,color%(HUIDA(x,y))
360     END FOR i
370   END FOR j
380 END DEFine rellena
390 :
400 DEFine FuNction color% (x%)
410   IF x%=1000 THEN RETurn (0)
420   IF x% > 300 THEN RETurn (7)
430   IF x% > 100 THEN RETurn (6)
440   IF x% > 20 THEN RETurn (((x% DIV 3) MOD 4) + 2)
450   RETurn (1)
460 END DEFine color%
470 :
480 DEFine PROCedure aumenta
490   x% = 0
500   y% = 0
510   dx% = 512
520   dy% = 256
530   OVER -1
540   box
550   REPeat loza
560     tecla=CODE(INKEY$(-1))
570     box
580     SELect ON tecla
590       = 192:IF x% > 0 THEN x%=x%-2
600       = 200:IF x% + dx% < 512 THEN x%=x%+2
610       = 216:IF y% + dy% < 256 THEN y%=y%+1
620       = 208:IF y% > 0 THEN y%=y%-1
630       = 193:IF dx% > 10 THEN dx%=dx%-2
640       = 201:IF dx% < 512 THEN dx%=dx%+2
650       = 217:IF dy% < 256 THEN dy%=dy%+1
660       = 209:IF dy% > 5 THEN dy%=dy%-1
670       = 10:nuevo:OVER 0:RETurn
680       = REMAINDER
690     END SELect
700   dx% = 512
710   dy% = 256
720   OVER -1
730   box
740   REPeat loza
750     tecla=CODE(INKEY$(-1))
760     box
770     SELect ON tecla
780       = 192:IF x% > 0 THEN x%=x%-2
790       = 200:IF x% + dx% < 512 THEN x%=x%+2
800       = 216:IF y% + dy% < 256 THEN y%=y%+1
810       = 208:IF y% > 0 THEN y%=y%-1
820       = 193:IF dx% > 10 THEN dx%=dx%-2
830       = 201:IF dx% < 512 THEN dx%=dx%+2
840       = 217:IF dy% < 256 THEN dy%=dy%+1
850       = 209:IF dy% > 5 THEN dy%=dy%-1
860       = 10:nuevo:OVER 0:RETurn
870       = REMAINDER
880     END SELect
890     box
900   END REPeat loza
910 END DEFine aumenta
920 :
930 DEFine PROCedure box
940   BLOCK dx%-1,1,x%,y%,7
950   BLOCK 2,dy%,x%+dx%-2,y%,7
960   BLOCK dx%-1,1,x%,y%+dy%-1,7
970   BLOCK 2,dy%,x%,y%,7
980 END DEFine box
990 :
1000 DEFine PROCedure nuevo
1010   ddx = dxi / 512
1020   ddy = dyi / 256
1030   XI = XI + x% * ddx
1040   yi = yi + (256-y%-dy%+1)*ddy
1050   dxi = dxi * dx% / 512
1060   dyi = dyi * dy% / 256
1070 END DEFine nuevo
```



Esta función, verdadera alma de los programas, a pesar de estar escrita en assembler no es una panacea universal, pues efectuar mil veces el elevar un número complejo al cuadrado, sumarle otro, guardar el resultado, hallar el cuadrado del módulo (más rápido que hallar el módulo) y compararlo con cuatro lleva un tiempo aproximado de tres segundo y medio, valor más que respetable si tenemos en cuenta que una pantalla con muchos puntos con valor 1.000 puede llegar a tardar dos días en calcularse completa punto a punto, cosa que hace el primer programa, pero que tendremos la seguridad de tener cada punto coloreado correctamente.

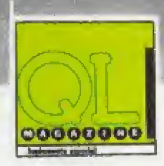
La descripción de este primer programa es la siguiente, para empezar pide como ya habíamos dicho la posición del pixel situado en la esquina inferior izquierda de la pantalla y los valores del ancho y alto de la pantalla, esto sirve para dar un factor de escala (parecido al comando SCALE del SuperBASIC, pero entre otras diferencias, introduciendo también el valor del ancho x. Con estos valores y el de la variable «factor» se procede al siguiente cálculo: como es conveniente trabajar con la instrucción BLOCK y no con la POINT, pues la BLOCK permite direccionar pixels directamente y la POINT no, se calcula el número de bloques que va a haber por línea y columna, dividiendo 512 ó 256 entre 2*factor y factor respectivamente (factor se aconseja que sea potencia de 2), cuando mayor sea factor, más baja resolución obtendremos, pero antes terminará el dibujo. Una vez obtenido el tamaño del bloque mínimo a dibujar, es necesario calcular el ancho y alto de éstos, pero referido esta vez al plano complejo donde estamos trabajando, para ello dividiremos el ancho y alto de la pantalla entre el número de bloques a dibujar respectivamente en horizontal y vertical. Trabajando ahora con dos bucles FOR anidados, vamos calculando los valores de la función HUIDA en cada punto considerado (nótese que para a cada bloque se le da el valor de su esquina inferior izquierda). Una vez obtenido este valor sólo hay que pasarlo por una función que nos devuelve un número entre 0 y 7, función a la que se debe la belleza de las imágenes que acompañan el artículo. Recomandamos para esta función que devuelva 0 sólo en el caso de que se le pase 1.000 como parámetro, dejando a la voluntad de los lectores el modificarla, pues dependiendo del aumento con el que estemos mirando el conjunto habrá que colorear de una forma y otra para obtener resultados más o menos interesantes, o para resaltar la estructura filamentosa del conjunto.

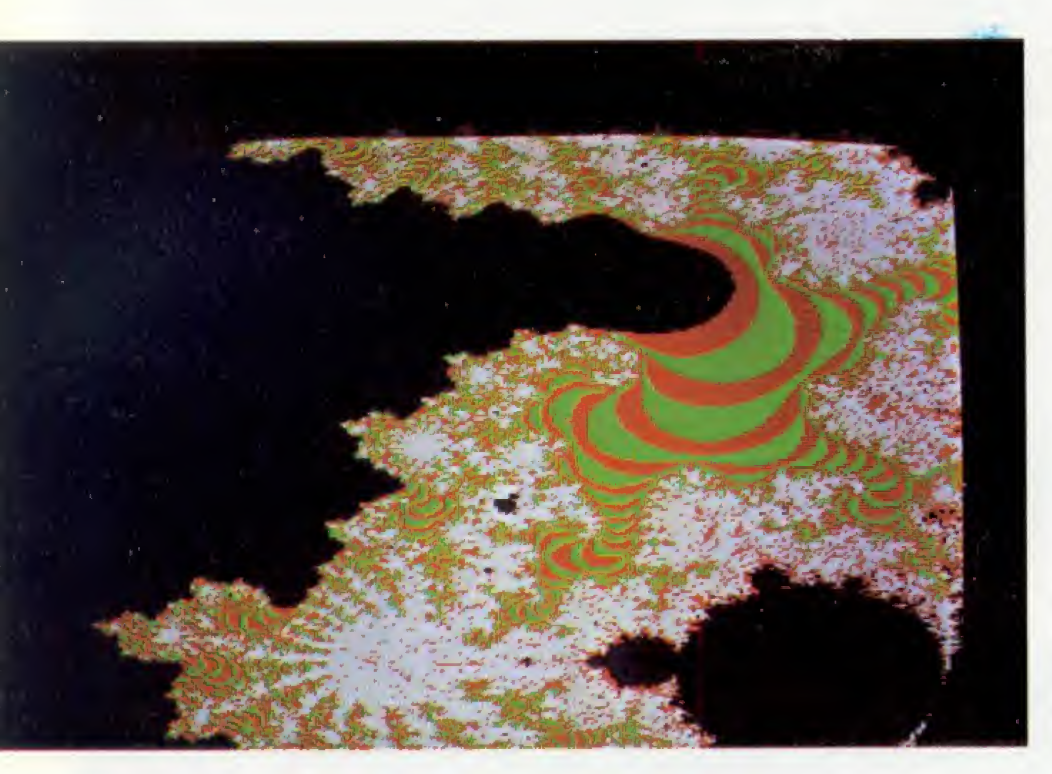
Para efectuar un trabajo más rápido, pero a costa de algunos errores, está el segundo programa que trabaja de forma recursiva.

Por los manuales del SuperBASIC, sabemos que se pueden escribir programas que sean recursivos. El significado de esta palabra es bastante oscuro y aterrador (algo parecido al código máquina) y representa la posibilidad de resolver un problema teniendo en cuenta que sabemos resolver el mismo problema pero con datos ligeramente más sencillos.

El problema en nuestro caso es pintar una columna de pixels, para ello seguimos el siguiente método: Obtenemos por medio de HUIDA el valor de los pixels superior e inferior de la columna, si tienen el mismo valor, coloreamos toda la co-

```
100 RESTORE
110 a=RESPR (256)
120 FOR i=0 TO 9
130   y=0
140   FOR j=0 TO 9
150     READ x
160     POKE_W a+i*20+j*2,x
170     y=y+x
180   END FOR j
190   READ x
200   IF y <> x THEN PRINT "Error en linea - ";1000+i*10
210 END FOR i
220 SBYTES mdv1_huida_cde,a,256
1000 DATA 17402,10,13432,272,20114,20085,0,0,1,12,71328
1010 DATA 1352,21833,17473,0,0,8764,0,60,13432,282,63196
1020 DATA 20114,13432,276,20114,3139,2,26112,96,17078,-26372,73991
1030 DATA 17078,-26376,17078,-26380,10313,-9732,0,12,-27652,0,-45659
1040 DATA 18,11708,16384,0,-26622,15804,2051,-26624,17028,21060,30807
1050 DATA 32256,18426,50,13432,286,20114,19072,26148,-11524,6,118266
1060 DATA 19126,-26372,27654,3140,1000,28126,28672,8780,21833,11593,123552
1070 DATA 88,15748,-26624,30723,20085,28913,20085,-6122,5646,-4586,83956
1080 DATA 3596,-3062,-5650,3606,2810,2582,-4330,3816,5646,2786,11800
1090 DATA 3072,0,0,0,0,0,0,0,0,3072
```





lumna del color correspondiente, si no dividimos la columna por el centro y formamos dos subcolumnas a las cuales damos el mismo tratamiento, y así ¿hasta cuando? Hasta que tengamos una columna de sólo dos pixels de alto, en cuyo caso coloreamos cada pixel con su color y seguimos a otra cosa.

Este método tiene dos inconvenientes, el primero es que cuando estamos tratando una columna en la que los pixels superior e inferior tienen valores distintos, al dividirla por el centro, el pixel correspondiente será siempre distinto de uno de los dos extremos, por lo que siempre se fuerza una serie de subdivisiones hasta llegar al caso de una columna de dos pixels; el otro problema al que hacíamos referencia es que cada vez que subdividimos una columna, es necesario recalcular los valores extremos de la columna original, siendo necesario en los casos extremos el hacer este cálculo hasta 8 veces (¿dónde está el ahorro?).

La solución a estos problemas viene dada en el programa que presentamos, el primero de ellos se soluciona, cuando al dividir una columna en dos partes no tomamos un solo punto intermedio, sino que la subcolumna superior es la delimitada por el pixel superior y central de la original, y la subcolumna inferior es la delimitada, no por el pixel central, sino por el inmediato inferior y el inferior de la original; el segundo problema se soluciona pasando al procedimiento recursivo que calcula el color de cada columna, no sólo las posiciones de los pixels extremos, sino también sus valores, con lo que al subdividir sólo tiene que calcular los valores de los nuevos puntos y llamarse a sí mismo recursivamente.

Los dos programas tiene al final unos procedimientos que se encargan, una vez terminada la pantalla, de permitir la ampliación de una zona de esta a voluntad; se consigue esto por medio de un rectángulo que aparece en la pantalla al terminarla, este rectángulo se mueve con las teclas de cursor y se vería su tamaño con la tecla ALT junto a las de cursor (más fácil es que probéis que explicarlo); una vez que tenemos la zona que nos interesa de la pantalla recuadrada, hay que dar «ENTER» con lo que se empezará a trazar ese recuadro ampliado a toda la pantalla.

La imagen de la portada es la representación de la zona del plano complejo que tiene como esquinas inferior izquierda −0.96+0.23i y−0.86+0.33i, por lo que habrá que introducir para verla x=−0.96 y=0.23 dx0.1 dy=0.1; la función color% necesaria para verla es la dada en el programa lento.

Si queréis ver el conjunto en su totalidad, probar con x=−2.1 y=−1.5 dx=3 dy=3, con lo que aparecerá una vista «a pájaro» del conjunto. Es muy curioso comprobar que en las proximidades del conjunto observamos con la suficiente ampliación copias miniatura de él, copias que no son exactamente iguales al original, pues un sorprendente teorema nos asegura que el conjunto es conexo, esto es que cada una de las reproducciones miniatura está unida al conjunto principal por un filamento más o menos grueso.

Otras vistas nos permiten vislumbrar espirales estilizadas filamentos aparentemente caóticos y electrizados y «ojos» que nos miran curiosos desde las profundidades de las matemáticas.

APENDICE

Un número complejo se define como suma de un número real y un número imaginario, los números imaginarios son aquellos que tienen como factor el número «i» o unidad imaginaria, este número es la raíz cuadrada de −1 y por lo tanto tiene como cuadrado −1, si intentamos obtener las potencias sucesivas de i, teniendo en cuenta que $i^2 = -1$, y que por definición $i^0 = 1$ tendremos:

potencia	0	1	2	3	4	5	6	7	8
ipotencia	1	i	-1	-i	1	i	-1	-i	1

Como vemos es un comportamiento bastante curioso.

Gerardo Izquierdo

Ordena tus propias ideas

Le sacarás partido a tu ordenador



TU PRIMER LIBRO DEL ZX SPECTRUM.
J. Dewhirst y R. Tennison
848 ptas.



"SPRITES" Y GRAFICOS EN LENGUAJE MAQUINA. (ZX SPECTRUM)
John Durst
1.537 ptas.



EL LIBRO GIGANTE DE LOS JUEGOS PARA ZX SPECTRUM
Tim Hartnell
1.431 ptas.



JUEGOS GRAFICOS DE AVENTURA. Técnicas de diseño.
Richard Hurley
1.484 ptas.



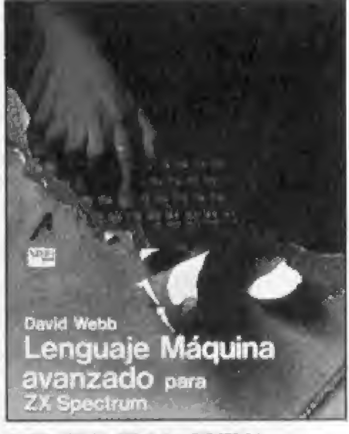
PROGRAMACION DEL INTERFACE I Y MICRODRIVE.
Agustín Núñez
1.166 ptas.



PROGRAMACION DEL Z80
Rodnay Zaks
2.915 ptas.



DISEÑO DE GRAFICOS Y VIDEOJUEGOS. TRATAMIENTO EN TRES DIMENSIONES.
(Incluye cassette)
Ian O. Angel y Brian J. Jones
3.392 ptas.



LENGUAJE MAQUINA AVANZADO PARA ZX SPECTRUM
David Webb
1.484 ptas.

☐ Les ruego me envíen el catálogo de su editorial.
☐ Les ruego me envíen los siguientes títulos:

____

____

TOTAL __________

☐ Adjunto talón bancario a GRUPO DISTRIBUIDOR EDITORIAL, S. A.
☐ Pagaré contrarrembolso (+ 125 pesetas de gasto de envío).
☐ Giro postal.

Nombre ____________________
Profesión ____________________
Dirección ____________________
C.P. __________ Localidad __________
Provincia ____________________
TS



Adquiéralos en su librería habitual.
Si no le es posible o desea que le enviemos nuestro catálogo, envíe este cupón a:
Apdo. de Correos 14632, Ref. D. de C. 28080 MADRID

ANAYA ANAYA ANAYA ANAYA ANAYA ANAYA

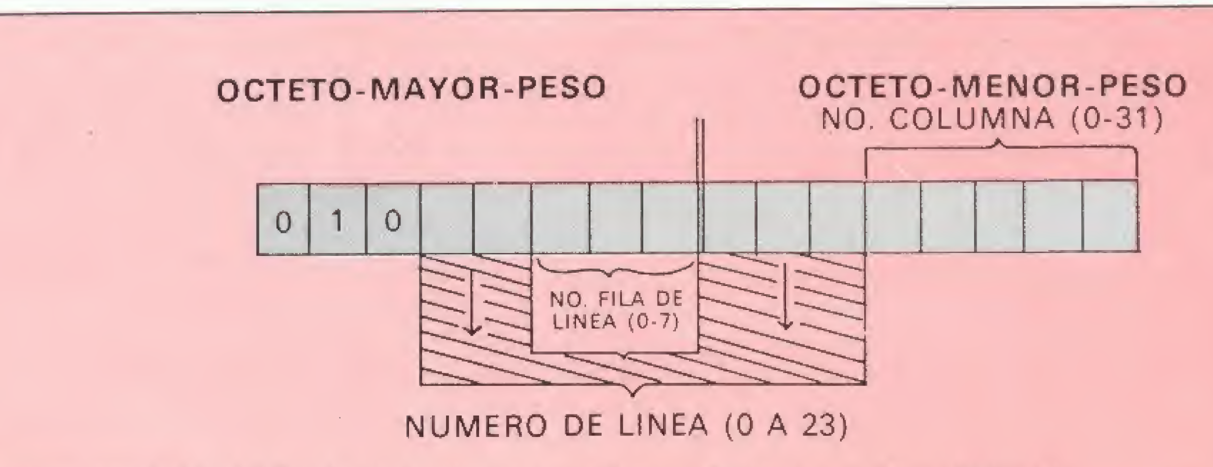
APREN
LENGUAJ

El microprocesador Z-80, aquel que estamos aprendiendo a programar, sólo es capaz de ver ante él un cierto número de direcciones a/ de las que puede enviar o tomar datos, y unos cuantos registros propios con los que puede manejar y operar estos datos y direcciones. Aparte de las operaciones de entrada y salida, que fueron vistas ya en el capítulo 7, en condiciones normales las mencionadas direcciones sólo pueden representar una cosa: posiciones de memoria. Es importante, por tanto, que tengamos una idea clara de la forma en que están distribuidas estas posiciones en nuestro querido Spectrum.

Como ya hemos visto en alguna ocasión, el Z-80, al tener un bus de direcciones de 16 bits, es capaz de direccionar hasta 64 kilobytes de memoria, es decir, hasta 65536 bytes. Esto es así en el Spectrum 48K, que aprovecha en su totalidad las posibilidades del Z-80 en este sentido, pero en el Spectrum 16K sólo la mitad de esta capacidad es utilizada (la forma en que el Spectrum 128K consigue sobrepasar este límite será tratada en otra ocasión).

Estas 64K que puede llegar a tener el Spectrum se dividen en tres partes principales:

16K de ROM, contenidas en un sólo chip, que es donde reside el sistema operativo, el intérprete BASIC, el juego de carácteres, los mensajes de error; es decir, todo aquello que necesita el Spectrum para funcionar y que debe estar ahí desde el



Ilustraciones reproducidas del libro *Lenguaje máquina avanzada para ZX-Spectrum*, de *Anaya Multimedia*.

DIENDO Capítulo 10

JE MAQUINA

momento en que se ponga en marcha.

16K de RAM dinámica, presentes tanto en el 16K como en el 48K, contenidas en 8 chips estándar 4116 de 16 Kbits (¡ojo!, Kbits, no Kbytes). Aquí es donde se encuentra «mapeado» el contenido de la pantalla en ese momento, así como la mayoría de la información variable que necesita el sistema y un cierto espacio de memoria libre para los programas del usuario.

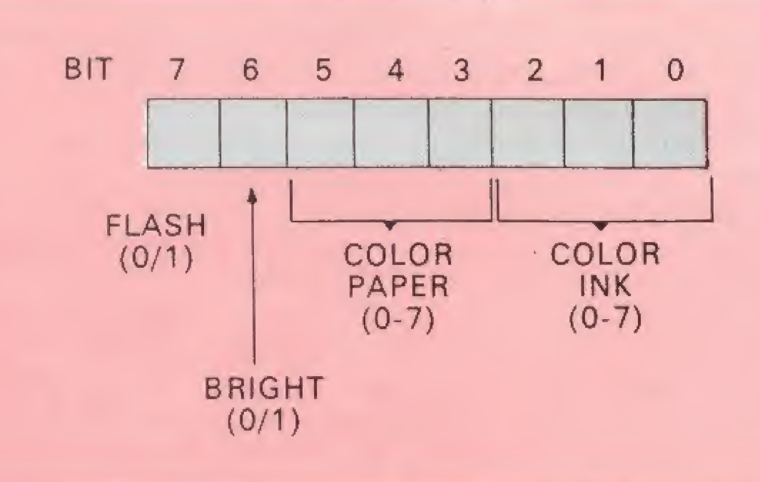
Las restantes 32K de RAM dinámica, sólo presentes en el 48K y contenidas en 8 chips 4532 de 32Kbits, tienen una curiosa peculiaridad, ya que son en realidad chips defectuosos de 64 Kbits. La razón de esto es que es muy difícil conseguir chips de memoria de 64 Kbits, y es frecuente que muchos de estos tengan algún defecto que los haga inútiles. Texas Instruments pensó en esto cuando diseñó sus chips de 64 Kbits de modo que pudieran funcionar también como dos mitades independientes de 32 Kbits; de esta forma cuando los defectos que hacen inútil un chip se limitan a una de estas dos partes, puede ser utilizado como 32 Kbit por creadores poco espléndidos como nuestro conocido Sinclair.

Como hemos visto, toda la memoria RAM utilizada en el Spectrum es dinámica; un tipo de memoria que, a diferencia de la estática que se puede mantener hasta que se desconecta el ordenador, necesita ser reescrita cada cierto (pequeño) intervalo de tiempo para que los datos se conserven. De este llamado «refresco» de la memoria se encargan, en el Spectrum, el Z-80 y la ULA actuando conjuntamente. A pesar de la aparente desventaja que supone esto, la memoria dinámica es la más utilizada en la actualidad, ya que pueden concentrarse más cantidad de bits en un solo chips a un precio mucho menor.

El Fichero de presentación visual

Si observamos detenidamente el mapa de memoria (sólo de la RAM) de la figura 1, podemos comprobar que las primeras 6K, una parte importante sin duda, corresponden al Fichero de presentación visual; pero, ¿qué queremos decir con eso de «fichero de presentación visual»?

En el Spectrum, como en muchos otros ordenadores, para almacenar en memoria lo que debe aparecer en la pantalla en cada momento se utiliza la técnica denominada «mapeado por bits». Consiste en dividir la pantalla en un cierto número de puntos (que en el Spectrum son 256*192, es decir 49152) y asignar un





bit a cada uno de estos puntos; de esta forma cuando un determinado bit tiene valor 1 aparece un punto en la zona de la pantalla que le corresponde; o sea, que a cada grupo de ocho puntos corresponde un byte en memoria, que es el que indica si esos puntos deben aparecer o no.

A diferencia de otros ordenadores, el Spectrum no posee un chip específico de vídeo que actualice la pantalla en cada momento según el modo de presentación que se esté utilizando. Esta tarea la realiza la ULA, con un único modo de presentación, leyendo cada cierto tiempo (50 veces por segundo) el contenido de la memoria de pantalla, generando las tres señales básicas del vídeo: B-Y y R-Y con la información de línea y cuadro, y enviando la señal correspondiente al modulador para que éste añada una portadora de muy alta frecuencia que le permita entrar por la antena del televisor como una emisora más.

Visto esto podemos comprender cómo, si seguimos los algoritmos adecuados, podemos dibujar en la pantalla simplemente cargando números en posiciones de memoria mediante la instrucción LD (lo mismo puede hacerse desde el BASIC con POKE). De todas formas no es nada fácil hacerse con el dominio de estas 6K, máxime si tenemos en cuenta la forma en que han sido distribuidas.

Toda la memoria RAM del Spectrum es dinámica y necesita ser refrescada cada cierto tiempo.

La forma que resultaría más cómoda a nuestras decimales mentes sería la de distribuir todos los bytes dando, de izquierda a derecha y de arriba a abajo, uno a cada grupo de ocho bits hasta terminar con la pantalla, pero si las cosas fueran tan fáciles no tendría ningún mérito programar en lenguaje máquina.

A la primera línea de 256 puntos le corresponden, como cabría esperar, los primeros 32 bytes (32*8=256) del «Fichero de presentación», sólo que los 32 bytes siguientes de memoria no representan a la segunda línea en pantalla sino a la novena (o lo que es lo mismo, a la primera de la segunda línea de carácteres). Este extraño orden sigue hasta completar el tercio superior de la pantalla (las primeras ocho líneas de carácteres), y entonces se comienza con el segundo tercio (como en los toros pero más pacíficamente). La forma más fácil de llegar a hacerse una idea del método utilizado es experimentar «pokeando» desde el BASIC.

El motivo principal que ha llevado a hacer una distribución como esta es que, al no llevar el Spectrum chip de vídeo, no se dispone de mo-

Todospectrum



TODOSPECTRUM es una publicación mensual que le ayudará a obtener el máximo partido a su **SPECTRUM** y al **ZX 81**.



CONOZCA LAS VENTAJAS DE SUSCRIBIRSE A

Todospectrum



ADEMAS, le hacemos un 25 % DE DESCUENTO

sobre el precio real de suscripción (12 números)

VALOR REAL DE SUSCRIPCION	OFERTA ESPECIAL DE SUSCRIPCION	USTED AHORRA
~~3.600~~ PTAS.	2.700 PTAS.	900 PTAS.

APROVECHE AHORA esta oportunidad irrepetible para suscribirse a **TODOSPECTRUM**. Envie **HOY MISMO** la tarjeta adjunta a la revista, que no necesita sobre ni franqueo. Deposítela en el buzón más cercano. Inmediatamente recibirá su primer ejemplar de **TODOSPECTRUM** más el **REGALO**.

Todospectrum
Bravo Murillo, 377
Tel. 733 79 69
28020 MADRID

El Z-80, con un bus de direcciones de 16 bits, puede direccionar un máximo de 64 kilobytes de memoria.

do texto como en otros ordenadores, por lo que se hacía necesario distribuir la pantalla de modo que se pudieran manejar fácilmente las posiciones de baja resolución (correspondientes no a puntos sino a casillas de carácteres) con el sistema hexadecimal que suelen utilizar los programadores de ensamblador. Así, para bajar una línea basta con incrementar el byte de más peso de la dirección, es decir, si el primer byte de un carácter está en la dirección 4000H el segundo estará en la 4100H, y el tercero en la 4200H.

Si pensamos en la pantalla como una cuadrícula de 24 por 32 carácteres y ocho filas por carácter podemos hacer la transformación de una coordenada cualquiera a la dirección equivalente en binario (o viceversa) de forma bastante sencilla (ver fig. 2): los bits 0 a 4 dan el número de columna entre 0 y 31, los bits 5, 6, 7, 11 y 12 dan el número de línea en baja resolución (entre 0 y 23), y los bits 8 a 10 dan el número de línea en alta resolución (entre 0 y 7), los tres bits más significativos son siempre 010. Sabiendo esto, y ayudados por las instrucciones lógicas, de rotación o de manipulación de bits, no es difícil hacer una subrutina que se ocupe de esta tarea.

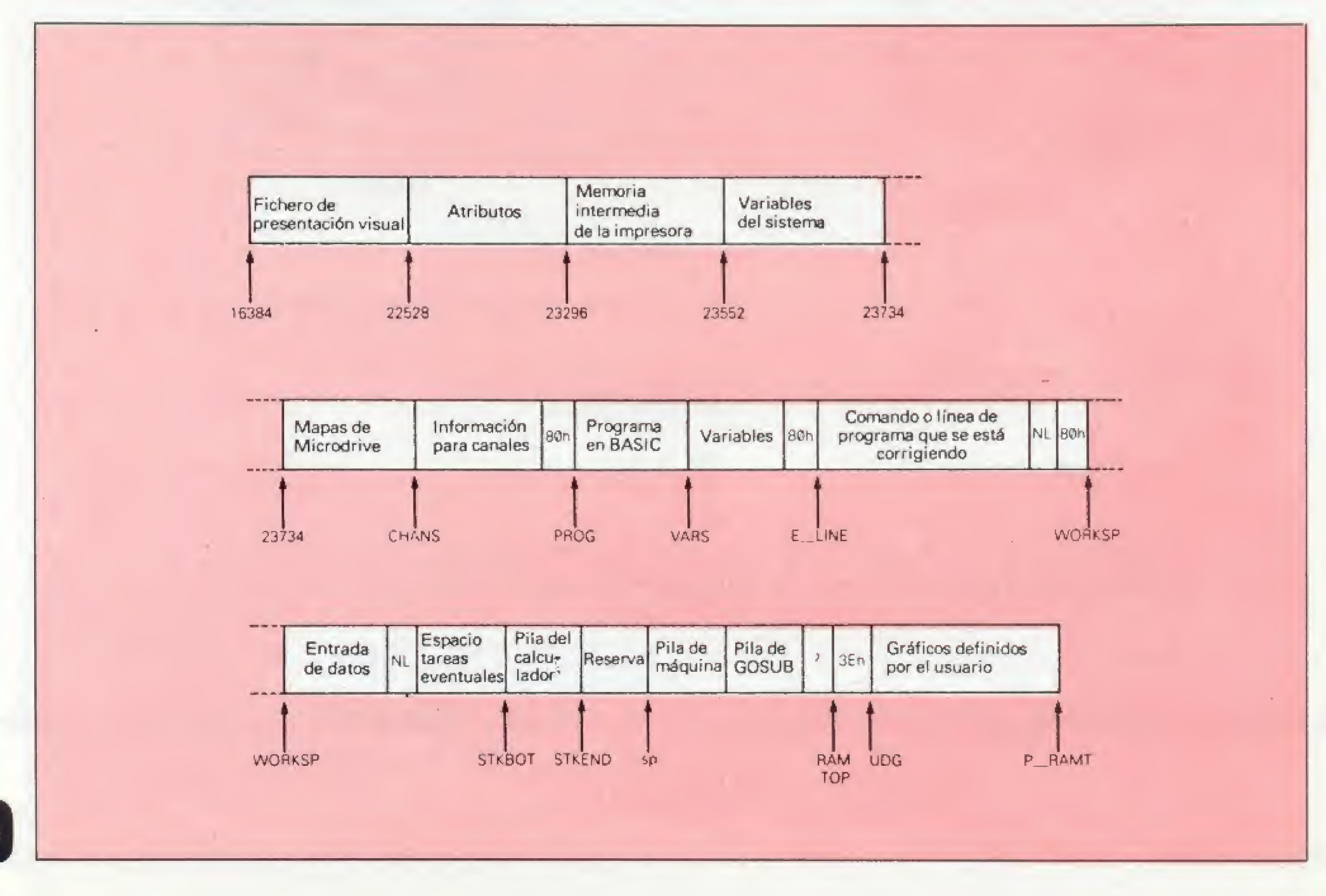
Démosle color

Lo expuesto anteriormente sobre la forma en que se las arregla el Spectrum para almacenar la información que en cada momento hay en pantalla puede entenderse medianamente bien si partimos de la base de que se utilizan sólo dos colores, pero no concuerda con las polícromas pantallas que podemos ver en todos los juegos comerciales. Sin embargo hay que resaltar que

El Spectrum no posee un clip específico de vídeo que actualice la pantalla. Esta tarea la realiza la ULA.

cada punto de la pantalla en el Spectrum sólo puede contener dos colores; el secreto está en que podemos elegir qué dos colores daremos a cada una de las celdillas de baja resolución (posiciones de carácter).

Hay una zona de memoria («Atributos» en el mapa de la figura 1) que se encarga de almacenar no sólo los colores que queremos dar a cada una de las 768 celdillas, sino también dos características más: el brillo y el parpadeo. A cada una de las posiciones le corresponde un byte, cuyos ocho bits tienen los cometidos siguientes: los tres primeros (b0 a b2) son el valor binario que corresponde al color de la tinta o INK (0= negro, 1=azul, etc.), del b3 al b5 se-



ñalan el color del fondo (PAPER), el bit 6 indica si hay brillo (BRIGHT), y el 7 si hay parpadeo o FLASH (ver fig. 3).

Puede hallarse fácilmente la dirección correspondiente a una casilla en el archivo de atributos usando la corta subrutina de la figura 4. A la entrada, el par HL debe tener las coordenadas en baja resolución; el registro H debe tener el número de línea (0-23) y L el de la columna (0-31). A la salida será el propio HL quien dé la dirección que le corresponda. Además de HL, sólo el acumulador resulta alterado.

Los 256 bytes del buffer de impresora sólo son utilizados cuando está conectada la ZX-Printer u otra impresora equivalente.

Si seguimos ascendiendo por el mapa de memoria nos encontramos con una zona denominada «Memoria intermedia (o Buffer) de la impresora». Estos son 256 bytes que normalmente sólo son utilizados cuando la ZX-Printer está conectada. Allí se almacena cada línea de 32 caracteres, no por sus códigos ASCII, sino como 8 grupos de 8 líneas cada uno. Si no tenemos la impresora conectada podemos utilizar este espacio para nuestras rutinas de código máquina o cualquier otra cosa, pero habremos de poner cuidado en no ejecutar ningún comando de la impresora (COPY, LLIST o LPRINT).

Las variables del sistema

Una de las zonas de memoria que mejor debe conocer el programador de lenguaje máquina es la correspondiente a las variables del sistema. En condiciones normales son 182 bytes donde se almacenan datos que indican el estado del sistema operativo en cada momento; allí se «apunta», por ejemplo, la posición donde se encuentra el cursor o los colores que están siendo utilizados, y sus «banderas» indican si se está editando una línea, se ejecuta un INPUT, etc.

En realidad si lo que proyectamos hacer es un programa en código máquina «puro», es decir, uno en el que no se utilice para nada las rutinas de la ROM y del que no se pretenda regresar al BASIC, no es necesario preocuparse en absoluto de estas variables, pero en la práctica sería de locos renunciar a todo un sistema operativo por querer aprovechardd esos pocos bytes.

Entre las variables más importantes se encuentran las que delimitan las distintas zonas de RAM, pues al tener muchas de estas zonas longitudes no fijas, van creciendo en la memoria «empujando» lo que tengan sobre sí. Así los nombres como CHANS, PROG, VARS, etc. que aparecen en el mapa de memoria indican que es en estas variables del sistema, o mejor, en las posiciones de memoria que les corresponden, donde podremos encontrar las direcciones que marcan el límite entre cada zona. Por ejemplo, si queremos saber la dirección de comienzo del programa BASIC, deberemos buscarlo en PROG, que, si buscamos en el manual, veremos que ocupa las posiciones 23635-6. Al ser un número de 16 bits es almacenado en la forma clásica, es decir, con el octeto menos significativo en primer lugar, por lo que si queremos imprimirlo en pantalla desde el BASIC deberemos hacer:

PRINT PEEK 23635+256*PEEK 23636

Las variables del sistema son 182 bytes que contienen información sobre el estado del sistema operativo en cada momento.

Las variables del sistema pueden ser utilizadas desde el código máquina de múltiples formas, al menos una para cada variable. Aunque es necesario conocer mínimamente el sistema operativo para poder utilizarlas correctamente (lo mejor es «empollarse» un poco el desensamblado de algunas zonas de la ROM), en el capítulo 25 del manual puede encontrarse una lista de todas ellas en la que se explica someramente su cometido y se da su dirección en decimal. Algunas de ellas pueden ser modificadas desde el BASIC para conseguir ciertos efectos, por ejemplo, CHARS, en la dirección 23606-7 marca el comienzo (menos 256) del juego de carácteres estándar; podemos modificar esta variable para hacerla apuntar a nuestro propio juego de carácteres y volver al anterior cuando queramos. PIP, en la dirección 23609, señala la longitud del chasquido que suena cada vez que pulsamos una tecla; podemos modificarlo para hacer que suene un pitido más audible, útil para controlar si ha aceptado una tecla o no.

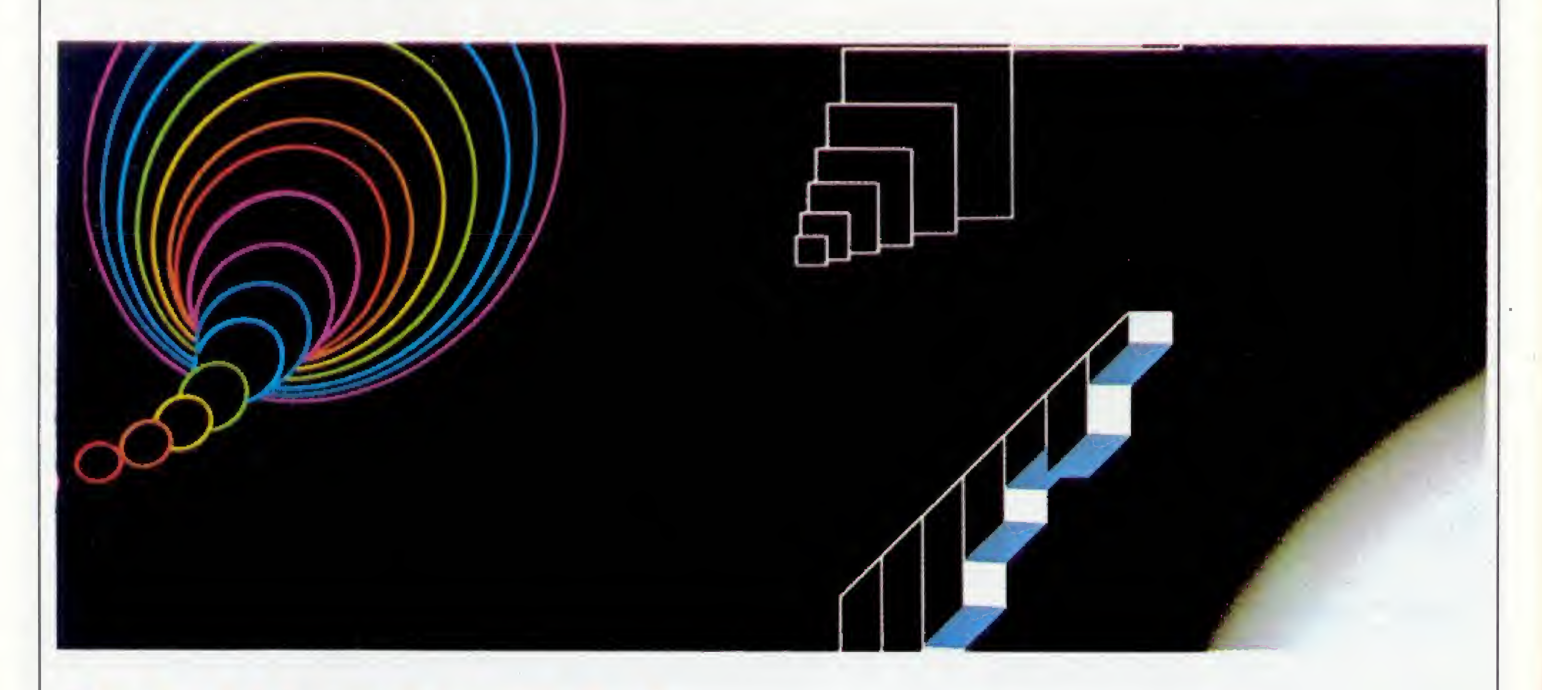
FIGURA 4

```
10  LC_ATR  LD    A,H
20          SRA   A
30          SRA   A
40          SRA   A
50          ADD   A,88
60          PUSH  AF
70          LD    A,7
80          AND   H
90          RRCA
100         RRCA
110         RRCA
120         ADD   A,L
130         POP   HL
140         LD    L,A
150         RET
```

En todo caso, y como ya se ha mencionado, lo importante es conocer bien cómo funciona cada cosa cuando el operativo está en marcha. Para ello la experimentación puede ser un buen arma cuando lo demás ha fallado. En el siguiente capítulo continuaremos con esta panorámica sobre el mapa de memoria del Spectrum para conseguir tener bien claro dónde pisamos en cada momento.

Luis Gala

CONICAS



Todos los elementos básicos de las curvas cónicas son tratados a fondo en los 22 Kbytes de memoria que ocupa este programa. Las curvas se pueden definir por fórmula o por condiciones, introduciendo cinco puntos o cuatro puntos y una tangente. Las instrucciones son completísimas y facilitan toda la información necesaria en cada fase del programa.

Definida la cónica, la pantalla mostrará su fórmula y el tipo de curva de que se trata (elipse, hipérbole, hipérbole equilátera o parábola). Pulsando una tecla aparece un extenso menú que cubre todas las posibilidades imaginables, desde polo de una recta a ecuación reducida de la cónica pasando por tangentes, asíntotas, vértices, representación gráfica e intersección recta - cónica.

```
  40 POKE 23609,150: PAPER 1: IN
K 7: BORDER 1: CLS : GO TO 700
  50 CLEAR : RESTORE : LET wv=0:
 LET ww=0: GO TO 1000
  70 CLS : GO TO 210
 101 LET wv=0: LET ww=0: BRIGHT
0: PAPER 1: INK 7: BEEP .5,20: P
RINT AT 18,8; FLASH 1;"PULSA UNA
 TECLA": PAUSE 30: PAUSE 0
 200 REM MENU
 210 CLS : PRINT "S     1- Polo d
e una recta .   E
                L     2- Polar
de un punto.Valor E      de la
conica para este. C   3- Centro
 de la conica     C"
 220 PRINT "I       4- Interseccion
 recta-          O     conica.
                 N  5- Ejes de la c
onica.           A
                    6- Asintotas (h
iperbola         U     unicamente).
                 N  7- Tangentes de
 un punto.       A"
 225 PRINT "        8- Vertices de
la conica.       O
                 P  9- Representaci
on grafica.      C"
 230 PRINT "I       R- Ecuacion red
ucida de la      O     conica.FOCOS
                 N  E- Definir otra
 conica."
 400 LET j$=INKEY$: IF j$="" THE
N GO TO 400
 410 LET o=2001*(j$="1")+1501*(j
$="2")+2501*(j$="3")+3001*(j$="4
")+3501*(j$="5")+4001*(j$="6")+4
501*(j$="7")+5001*(j$="8")+5501*
(j$="9")+2201*(j$="r")+10*(j$="e
")
```

```
 420 GO TO o+400*(o=0)
 700 REM PRESENTACION
 705 PRINT AT 15,0;"  GUILLERM
O CANDELA   1984": PRINT : PRINT
 : PRINT "          PULSA UNA TECL
A"
 710 FOR n=0 TO 31: PRINT AT 7,3
1-n; INK n/7+2;"



                "
 720 IF INKEY$<>"" THEN  GO TO 5
0
 730 BEEP .008,n: NEXT n: GO TO
710
1000 REM INTRODUCCION CONICA
1002 PRINT "   Puedes definir un
a conica porsu formula o por con
diciones:    Elige una opcion":
PRINT : PRINT "    1- POR SU FORM
ULA.": PRINT : PRINT "    2- POR
CONDICIONES."
1003 IF INKEY$="1" THEN  CLS : G
O TO 1007
1004 IF INKEY$="2" THEN  GO TO 6
000
1005 GO TO 1003
1007 PRINT "        La representa
cion generalde una conica viene
dada por la siguiente expresion:

            Ax^2 + By^2 + Cxy +
Dx + Ey + F
                    Necesitamos s
aber los     valores de A,B,C,D,E
,F. Caso de ser nulos,pon el val
or 0."
1010 INPUT "Introduce los valore
s: A=";k;" B=";b;"   C=";c;" D="
;d;" E=";e;" F=";f
1090 RESTORE 1090: READ aa,ab,ac
,dis: DATA (k*b-(c*c)/4),-((d/2)
*b-(c*e)/4),((d*c)/4-(k*e)/2),(a
a*f+(ab*d)/2+(ac*e)/2)
1095 DEF FN c(x,y)=k*x*x+b*y*y+c
*y*x+d*x+e*y+f
1100 REM CLASIFICACION CONICA
1110 IF dis=0 THEN  GO TO 1300
1120 IF aa>0 THEN  LET a$="ELIPS
E"
1125 IF aa>0 AND k=b AND (c/2)=0
 THEN  LET a$="CIRCUNFERENCIA"
1127 IF aa>0 AND k*dis>0 THEN  L
ET a$=a$+" IMAGINARIA"
1130 IF aa<0 THEN  LET a$="HIPER
BOLA"
1135 IF aa<0 AND k+b=0 THEN  LET
 a$=a$+" EQUILATERA"
1140 IF aa=0 THEN  LET a$="PARAB
OLA"
1200 CLS : PRINT "   La conica:"
,,,,: PRINT k;"x^2 +",,b;"y^2 +"
,,c;"xy +",,d;"x +",,e;"y +",,f,
,
1210 PRINT "",,"es una",,,,a$
1250 GO TO 101
1300 CLS : PRINT ,,"   ESTA CONI
CA ES DEGENERADA",,,"ES,PUES,UNA
 O DOS RECTAS REALES           O
IMAGINARIAS"
1350 GO TO 101
1500 REM Polar
1510 CLS : INPUT "Introduce un p
unto definido por coordenadas ho
mogeneas P(c,x,y);sabiendo que (
c) es coordenada  del punto del
```

```
    Hay dos conicas con estas
 condiciones: PRIMERA
 Estas condiciones definen la
conica:
1 x↑2 +
336.18988 y↑2 +
-414.87815 xy +
2036.3907 x +
-22289.174 y +
103177.12  = 0
 SEGUNDA
 Estas condiciones definen la
conica:
1 x↑2 +
4.9533035 y↑2 +
8.7965498 xy +
-81.982749 x +
-373.63756 y +
1880.3552  = 0

Escoge una: Pulsa 1 o 2
```

```
 -5.0465561,7.5293993
ESC=24.4
RES=3

 -5.0465561,0.39825178
```

```
infinito, y que   toma el valor
 '1' si el punto esreal o el '0'
si es del infinitoindependientem
ente de X e Y, lastipicas coorde
nadas en el plano:
                   c=";u;" x=";x
;" y=";y
1520 RESTORE 1520: READ l,m,n,va
l: DATA u*d/2+k*x+(c/2)*y,u*e/2+
(c/2)*x+b*y,u*f+(d/2)*x+(e/2)*y,
k*x*x+b*y*y+c*y*x+d*x*u+e*u*y+f*
u*u
1525 IF ww=1 THEN  GO TO 3013
1528 PRINT "   EL VALOR DE LA CO
NICA EN ESTE PUNTO ES:
               F(x)=";val: PRINT
1530 IF m<>0 THEN  LET l=-l/m: L
ET n=-n/m: LET m=-1
1535 PRINT "  LA POLAR DE P(";u;
",";x;",";y;") ES:": PRINT "",,"
(";l;")x +(";m;")y +(";n;")"
1560 GO TO 101
2000 REM POLO DE UNA RECTA
2010 CLS : IF dis=0 THEN  PRINT
"      ESTA CONICA ES DEGENERADA;
 PARA CADA RECTA HAY INFINITOS
 POLOS.": GO TO 101
2020 INPUT "      Introduce la re
cta de la   cual quieres saber e
l polo de   la siguiente forma:

                  Ax + By + C
= 0
              A=";l;"  B=";m;"
C=";n
2030 LET u=(n*k*b+l*c*e/4+m*d*c/
4)-(e*k*m/2+c*c*n/4+b*d*l/2)
2035 LET x=(l*f*b+m*d*e/4+n*e*c/
4)-(e*e*l/4+c*f*m/2+b*d*n/2)
2040 LET y=(m*f*k+n*d*c/4+l*e*d/
4)-(k*e*n/2+c*f*l/2+m*d*d/4)
2045 IF u=0 THEN  PRINT "      El
 polo de esta recta es unpunto d
el infinito; pero puede  darse e
n coordenadas homogeneas:
                           POLO=
(0,";x;",";y;")": GO TO 101
2050 PRINT "       EL POLO DE LA R
ECTA          ": PRINT "(";l;")x
+ (";m;")y + (";n;")=0": PRINT :
 PRINT " ES EL PUNTO :": PRINT :
 PRINT " P(";x/u;",";y/u;")": GO
 TO 101
2200 REM FORMA REDUCIDA. FOCOS
2205 LET ww=3: CLS : IF dis=0 TH
EN  PRINT "      ESTA CONICA ES D
EGENERADA YNO SE PUEDE EXPRESAR
EN FORMA   REDUCIDA": GO TO 101
2210 IF aa=0 THEN  GO TO 2300
2213 IF aa>0 THEN  LET a$="ELIPS
E"
2216 IF aa<0 THEN  LET a$="HIPER
BOLA"
2220 RESTORE 2220: READ a,g,h,i,
h,t,h: DATA dis/aa,(k+b),SQR (g*
g-aa*4),(g-h)/2,(g+h)/2,-a/i,-a/
h
2230 PRINT "       La expresion re
ducida de   esta ";a$;" es la si
guiente:": PRINT : PRINT " x^2/(
";t;") + y^2/(";h;") = 1": LET w
=SQR (ABS (t-h))
2235 PRINT : PRINT : PRINT "  PU
LSA UNA TECLA PARA VER LOS    FO
COS": BEEP .4,15: PAUSE 0: GO TO
 5000
2240 LET x=(sx-sz): LET y=(sy-sw
): LET z=(x*x+y*y)/4: IF aa>0 AN
D z<ABS (t)*0.9991 THEN  GO TO 5
240
2250 RESTORE 2250: READ a,y,z,h,
x: DATA (ATN (1))*(m<>0)+(m=0)*P
I/2,w*SIN (a)+ac/aa,-w*SIN (a)+a
c/aa,w*COS (a)+ab/aa,-w*COS (a)+
ab/aa: PRINT : PRINT "      LOS F
OCOS SON LOS PUNTOS:": PRINT : P
RINT "F1(";h;",";y;")": PRINT "F
2(";x;",";z;")": GO TO 101
2260 RESTORE 2260: READ a,g,h,x,
y,a,g,h,i,u,v: DATA (-k*e+b*(e-d
)/2+c*(d+e)/4),((d*(d-e)-e*e)/4+
f*(b-k+c/2)),(c*c/4-b*k)
2265 DATA (-a+SQR (a*a-4*g*h))/(
2*h),(-a-SQR (a*a-4*g*h))/(2*h),
(e*c/4-d*b/2),(f*c/2-d*e/4),k*b-
c*c/4,k*e/2-c*d/4
2270 DATA (a*x+g)/(h*x+i),(a*y+g
)/(h*y+i)
2280 PRINT x;",";u: PRINT y;",";
v: STOP
2300 LET a=(k+b): LET h=SQR (-di
s/a): LET s=h/(2*a): PRINT "
 La expresion reducida de   esta
 PARABOLA es la siguiente:": PRI
NT : PRINT "  y^2 + (";2*h/a;")x
 = 0": PRINT : PRINT "siendo el
parametro=";s*2
2305 PRINT AT 16,7;"PULSA UNA TE
CLA": BEEP .4,17: PAUSE 0
2310 CLS : GO TO 5400
2320 IF l=0 THEN  LET g=sx+s: LE
T i=sy: IF FN c(g,i)<0 THEN  GO
```

```
TO 2400
2325 IF l=0 THEN  LET g=sx-s: GO
 TO 2400
2330 IF m=0 THEN  LET g=sx: LET
i=sy+s: IF FN c(g,i)<0 THEN  GO
TO 2400
2335 IF m=0 THEN  LET i=sy-s: GO
 TO 2400
2340 LET a=ATN (-l/m): LET g=sx+
s*COS (a): LET i=sy+s*SIN (a): I
F FN c(g,i)<0 THEN  GO TO 2400
2350 LET g=sx-s*COS (a): LET i=s
y-s*SIN (a)
2400 PRINT "      El FOCO de esta
 PARABOLA esel punto:
               F(";g;",";i;")": G
O TO 101
2500 REM CENTRO
2510 CLS : IF aa=0 THEN  PRINT "
     El CENTRO de esta parabola
es un punto del infinito. Sin
embargo lo podemos expresar en
COORDENADAS HOMOGENEAS:
 ": IF ab<>0 THEN  PRINT "  C=(0
,1,";ac/ab;")": GO TO 2550
2515 IF aa=0 THEN  PRINT "  C=(0
,0,";ac;")": GO TO 101
2520 PRINT "      EL CENTRO DE ES
TA CONICA ES",,: PRINT "  C=(";a
b/aa;",";ac/aa;")"
2550 GO TO 101
3000 REM INTERSECCION RECTA-CONI
CA
3010 CLS : INPUT "      Introduce
 la RECTA de la  siguiente mane
ra:


                           ax+by
+c=0
                  siendo a=";l;"
 b=";m;" c=";n
3013 IF l=0 THEN  GO TO 3200
3016 IF m=0 THEN  GO TO 3250
3020 RESTORE 3020: READ x,y,z,a:
 DATA ((k*m*m)/(l*l)-(c*m)/l+b),
((2*k*n*m)/(l*l)-(c*n+d*m)/l+e),
((k*n*n)/(l*l)-(d*n)/l+f),y*y-4*
x*z
3025 IF a<=0 THEN  GO TO 3100
3028 IF x<10^-7 THEN  LET sy=-z/
y: LET sx=-(n+m*sy)/l: GO TO 316
0
3030 RESTORE 3030: READ g,h,sx,s
y,sz,sw: DATA (-y+SQR (a))/(2*x)
,(-y-SQR (a))/(2*x),-(n+m*g)/l,g
,-(n+m*h)/l,h
3035 IF wv=1 OR wv=2 THEN  GO TO
 5200
3036 IF ww=1 THEN  GO TO 4600
3038 PRINT "      LOS PUNTOS DE C
ORTE DE LA  RECTA CON LA CONICA
SON:
               P(";sx;",";sy;")"
3040 PRINT : PRINT "  Q(";sz;","
;sw;")": GO TO 101
3100 IF a=0 THEN  LET sx=-(n+m*(
(-y)/(2*x)))/l: LET sy=(-y)/(2*x
): GO TO 3160
3150 IF ww=1 THEN  GO TO 4650
3152 IF wv=1 THEN  GO TO 5300
3155 PRINT "      LOS CORTES DE E
STA RECTA  CON LA CONICA SON IM
AGINARIOS": GO TO 101
```



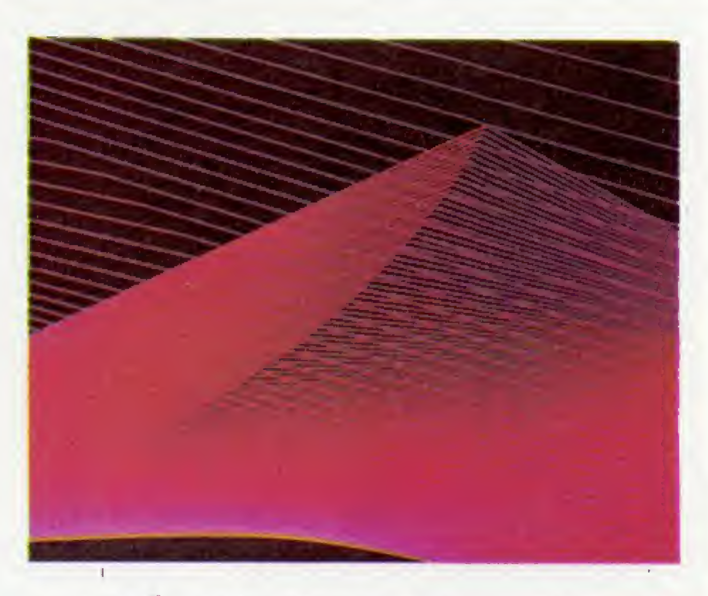
```
3160 IF ww=1 THEN  GO TO 4700
3163 IF ww=3 THEN  GO TO 2320
3165 IF wv=1 THEN  GO TO 5480
3168 IF wv=4 THEN  GO TO 5850
3170 PRINT "      EL UNICO PUNTO
DE CORTE ES:
               P(";sx;",";sy;")":
GO TO 101
3200 LET i=-(n/m): LET y=c*i+d:
LET z=f+b*i*i+e*i: LET a=y*y-4*z
*k: IF a<0 THEN  GO TO 3150
3210 IF a=0 THEN  LET sx=-(y/(2*
k)): LET sy=i: GO TO 3160
3220 LET sx=(-y+SQR a)/(2*k): LE
T sy=i: LET sw=i: LET sz=(-y-SQR
 a)/(2*k): GO TO 3035
3250 LET i=-(n/l): LET y=c*i+e:
LET z=f+k*i*i+d*i: LET a=y*y-4*z
*b: IF a<0 THEN  GO TO 3150
3260 IF a=0 THEN  LET sy=-(y/(2*
b)): LET sx=i: GO TO 3160
3270 LET sy=(-y+SQR a)/(2*b): LE
T sx=i: LET sz=i: LET sw=(-y-SQR
 a)/(2*b): GO TO 3035
3500 REM EJES DE LA CONICA
3510 CLS : IF aa=0 THEN  GO TO 3
700
3515 IF c=0 AND b<>k THEN  PRINT
 "   LOS EJES DE ESTA CONICA SON:
": PRINT : PRINT "x=";ab/aa;"  y
=";ac/aa : GO TO 101
3516 IF c=0 THEN  PRINT "      Es
ta conica es una         CIRCUN
FERENCIA; por ello todo  par de
 diametros perpendicula- -res e
ntre si se pueden tomar  como e
jes.": GO TO 101
3520 RESTORE 3520: READ g,x,y,z,
ex,ey,dx,dy: DATA (k-b),g*g+c*c,
(-g+SQR x)/c,(-g-SQR x)/c,y,ac/a
a-(ab*y)/aa,z,ac/aa-(ab*z)/aa
3530 IF wv=1 THEN  GO TO 5045
3535 IF wv=3 THEN  GO TO 5540
3550 CLS : PRINT "      LOS EJES
DE ESTA CONICA SONLAS DOS SIGUIE
NTES RECTAS:
                   (";ex;")x - y
 +(";ey;")": PRINT : PRINT " (";
dx;")x - y +(";dy;")": GO TO 101
3700 CLS : IF c=0 AND k=0 THEN
PRINT "       EL EJE ES LA RECTA:
          y=";(-e)/(2*b): GO TO 10
1
3710 IF c=0 AND b=0 THEN  PRINT
"      EL EJE ES LA RECTA:
  x=";(-d)/(2*k): GO TO 101
3720 PRINT "         EL EJE ES LA RE
CTA:          ": PRINT "(";k+b;")
x +(";c/2+(2*b*b)/c;")y +(";d/2+
e*b/c;")": GO TO 101
4000 REM ASINTOTAS HIPERBOLA
4010 CLS : IF aa>=0 OR dis=0 THE
N PRINT "       ESTA CONICA TIENE
 LAS        ASINTOTAS IMAGINARIAS
": GO TO 101
4020 LET a=c*c-4*k*b: IF b<>0 TH
EN  LET y=(-c+SQR a)/(2*b): LET
z=(-c-SQR a)/(2*b): PRINT "
LAS ASINTOTAS DE ESTA       HIPER
BOLA SON LAS RECTAS:

                             (";y;
")x - y +(";ac/aa-y*ab/aa;")": P
RINT : PRINT "(";z;")x - y +(";a
c/aa-z*ab/aa;")": PRINT : PRINT
: PRINT "   QUE FORMAN UN ANGULO
 DE ",,,ABS (ATN y-ATN z)*(180/P
I);" GRADOS": GO TO 101
4100 IF k<>0 THEN  LET y=(-c+SQR
 a)/(2*k): LET z=(-c-SQR a)/(2*k
): PRINT "        LAS ASINTOTAS DE
ESTA        HIPERBOLA SON LAS RECT
AS:

            x -(";y;")y +(";ac*y/a
a-ab/aa;")": PRINT : PRINT "x -(
";z;")y +(";ac*z/aa-ab/aa;")": P
RINT : PRINT : PRINT "     QUE FOR
```

```
MAN UN ANGULO DE ",,,ABS (ATN y-
ATN z)*(180/PI);" GRADOS": GO T
O 101
4200 PRINT "      LAS ASINTOTAS D
E ESTA       HIPERBOLA SON LAS RE
CTAS:
                 x=";ab/aa,"y=";ac/
aa: GO TO 101
4500 REM TANGENTES DE UN PUNTO
4510 CLS : INPUT "       INTRODUCE
 EL PUNTO: P(x,y)
                 x=";x;" , y=";
y
4515 IF FN c(x,y)=0 THEN  GO TO
4700
4520 LET s=x: LET t=y: LET u=1:
LET ww=1: GO TO 1520
4600 CLS : PRINT "     LAS TANGE
NTES SON :": PRINT : PRINT
4610 PRINT " (";(sy-t)/(sx-s);")
x - y + (";t-((sy-t)/(sx-s))*s;"
)": PRINT : PRINT "PUNTO DE TANG
ENCIA:               ": PRINT "
P(";sx;",";sy;")": PRINT : PRINT

4630 PRINT " (";(sw-t)/(sz-s);")
x - y + (";t-((sw-t)/(sz-s))*s;"
)": PRINT : PRINT "PUNTO DE TANG
ENCIA:               ": PRINT "
P(";sz;",";sw;")": GO TO 101
4650 PRINT "       Desde este punt
o NO se     pueden trazar TANGEN
TES A LA     CONICA (punto interi
or...)": GO TO 101
4700 PRINT "    Este punto perten
ece a la     conica; solo tiene u
na tangente:su POLAR:": LET ww=0
: GO TO 1520
5000 REM VERTICES
5005 LET wv=1: IF aa=0 THEN  GO
TO 5400
5010 CLS : IF c=0 AND b<>k THEN
 GO TO 5050
5020 IF c=0 THEN  PRINT "     Est
a conica es una          CIRCUN
FERENCIA; por ello, no    tiene
vertices": GO TO 101
5040 GO TO 3520
5045 LET l=ex: LET m=-1: LET n=e
y: GO TO 3013
5050 LET n=-ab/aa: LET m=0: LET
l=1: GO TO 3013
5200 IF aa>0 THEN  LET a$="ELIPS
E": GO TO 5220
5210 LET a$="HIPERBOLA"
5220 IF wv=2 THEN  GO TO 5260
5225 IF wv<>2 AND ww=3 THEN  GO
TO 2240
5230 PRINT "     LOS VERTICES DE
ESTA           ";a$;" SON:": PRINT
```



```
: PRINT "V1=(";sx;",";sy;")": PR
INT "V2=(";sz;",";sw;")"
5240 IF aa>0 THEN  LET wv=2: GO
TO 5300
5250 GO TO 101
5260 IF ww=3 THEN  GO TO 2240
5270 PRINT : PRINT "V3=(";sx;","
;sy;")": PRINT "V4=(";sz;",";sw;
")": GO TO 101
5300 IF c=0 THEN  LET n=-ac/aa:
LET l=0: LET m=1: GO TO 3013
5305 LET l=dx: LET m=-1: LET n=d
y: GO TO 3013
5400 IF c=0 AND b=0 THEN  LET sx
=(-d)/(2*k): LET sy=(-(d*d/(4*k)
)+f)/(-e): LET l=1: LET m=0: LET
 n=-sx: GO TO 5480
5405 IF c=0 AND k=0 THEN  LET sy
=(-e)/(2*b): LET sx=(-(e*e/(4*b)
)+f)/(-d): LET l=0: LET m=1: LET
 n=-sy: GO TO 5480
5410 LET l=(k+b): LET m=(c/2+2*b
*b/c): LET n=(d/2+e*b/c): GO TO
3013
5480 IF wv=4 THEN  GO TO 5850
5482 IF ww=3 THEN  GO TO 2320
5485 CLS : PRINT "    EL VERTICE
DE ESTA PARABOLA  ES EL PUNTO:":
 PRINT : PRINT "V=(";sx;",";sy;"
)": GO TO 101
5500 REM REPRESENTACION GRAFICA
5501 CLS : PRINT "        Lee esta
s instrucciones   antes de segui
r: cuando pulses  una tecla se e
mpezara a dibujar la conica con
```

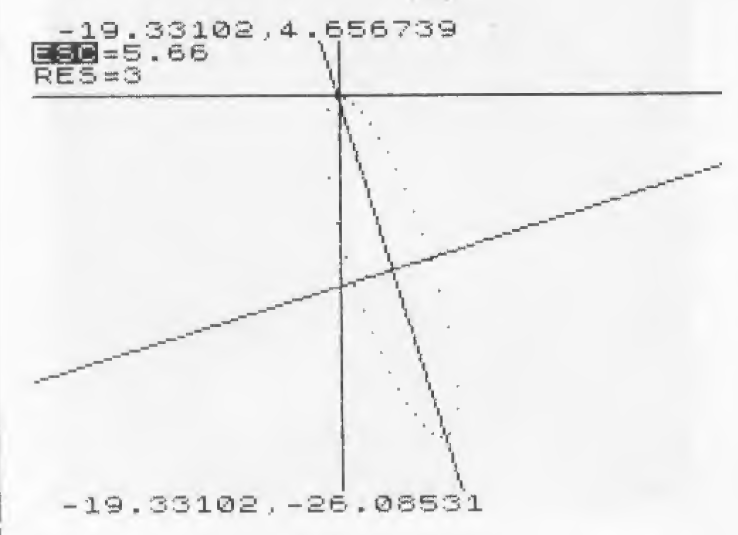
GUSANEZ

por José C. Tomás







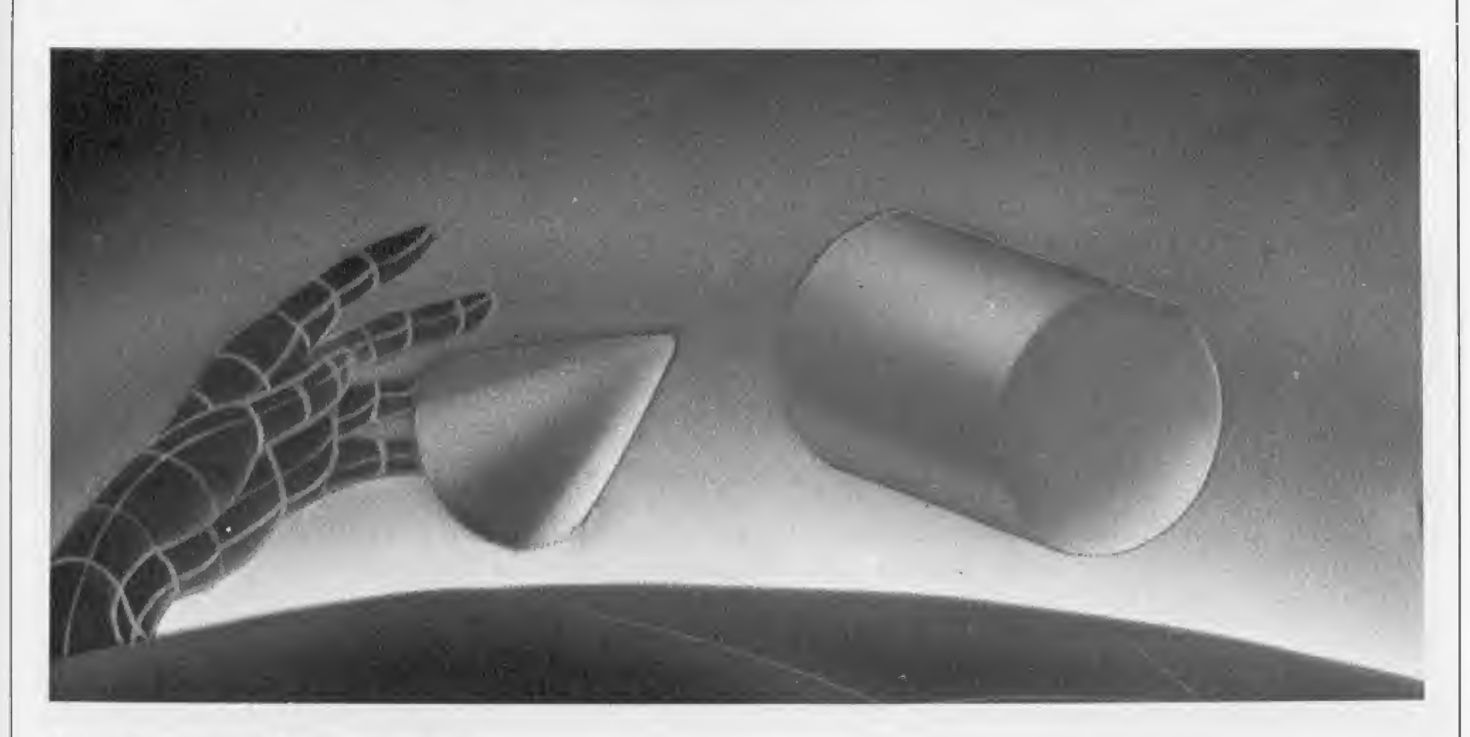


SELECCIONA UNA OPCION

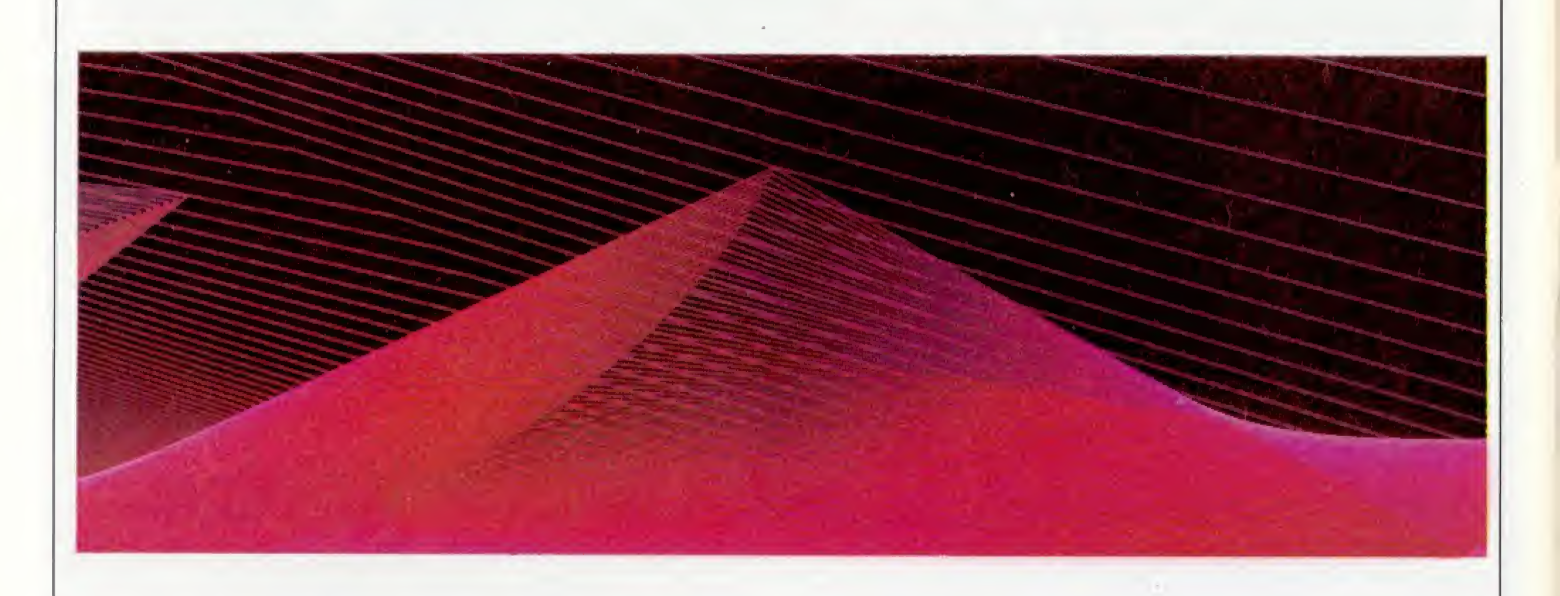
- 1- Polo de una recta .
- 2- Polar de un punto.Valor de la conica para este.
- 3- Centro de la conica
- 4- Interseccion recta-conica.
- 5- Ejes de la conica.
- 6- Asintotas (hiperbola unicamente).
- 7- Tangentes de un punto.
- 8- Vertices de la conica.
- 9- Representacion grafica.
- R- Ecuacion reducida de la conica.FOCOS
- E- Definir otra conica.

```
escala automatica,resolucion 3 y
 sin relleno.      Pulsa 'P' si q
uires que deje de dibujar, o bie
n esperar hasta elfinal. Te apar
ecera abajo 'Pulsatecla'y las te
clas de opciones:": PRINT : PRIN
T " A- Cambio de escala,resoluci
on  ": PRINT " M- Vuelta al MENU
 principal.    ": PRINT " C- Cop
ia de la pantalla en la      imp
resora                   S- Gra
ba la pantalla en cinta": PRINT
: PRINT "         PULSA UNA TECLA
": PAUSE 0
5502 INK 0: PAPER 5: BRIGHT 1: C
LS : LET p$="x": IF dis=0 AND aa
<>0 THEN  LET r=1: LET o=2: GO T
O 5515
5503 IF dis=0 THEN  PRINT "CONIC
A DEGENERADA:1 o 2 rectas  paral
elas": GO TO 101
5504 IF aa=0 THEN  LET a=k+b: LE
T h=SQR (-dis/a): LET r=(INT (40
00*a/h))/100: LET o=2: GO TO 550
6
5505 RESTORE 5502: READ a,g,h,i,
t,r,o: DATA dis/aa,(k+b),SQR (g*
g-aa*4),(g-h)/2,SQR ABS (-a/i),(
INT (63007t))/100,3
5506 BEEP .3,15: PRINT AT 1,0;"E
SC=";r: PRINT AT 2,0; PAPER 4;"R
ES"; PAPER 5;"=";o: GO TO 5515*(
aa<>0)+5800*(aa=0)
5510 INPUT "Introduce la ESCALA=
";r: INPUT "Introduce la RESOLUC
ION=";o: INPUT "Quieres RELLENO
? (s/n)";p$: BEEP .3,15: BRIGHT
1: PAPER 5: INK 0: CLS : PRINT A
T 1,0;"ESC=";r: PRINT AT 2,0; PA
PER 4;"RES"; PAPER 5;"=";o: IF a
a=0 THEN  GO TO 5800
5515 LET h=ab/aa: LET i=ac/aa
5520 IF c=0 AND b<>k THEN  PLOT
127,0: DRAW 0,175: PLOT 0,87: DR
AW 255,0: GO TO 5550
5525 IF c=0 THEN  GO TO 5550
5530 LET wv=3: GO TO 3520
5540 IF ex*(h-127)+ey>=(i-87) AN
D ex*(h-127)+ey<(i+87) THEN  PLO
T 0,ex*(h-127)+ey-i+87: DRAW 255
,ex*255
5543 IF ex*(h-127)+ey<(i-87) OR
ex*(h-127)+ey>=(i+87) THEN  PLOT
 (i-87-ey)/ex-h+127,0: DRAW 175/
ex,175
5545 IF dx*(h-127)+dy>=(i-87) AN
D dx*(h-127)+dy<(i+87) THEN  PLO
T 0,dx*(h-127)+dy-i+87: DRAW 255
,dx*255
5548 IF dx*(h-127)+dy<(i-87) OR
dx*(h-127)+dy>=(i+87) THEN  PLOT
 (i-87-dy)/dx-h+127,0: DRAW 175/
dx,175
5550 IF -i<=87/r AND -i>=-87/r T
HEN  PLOT 0,87-i*r: DRAW  INK 2;
255,0
5555 IF -h<=127/r AND -h>=-127/r
 THEN  PLOT 127-h*r,0: DRAW  INK
 2;0,175
5560 PLOT 0,0: PLOT 0,175: PRINT
 AT 0,1;(h-127/r);",";(87/r+i):
PRINT AT 21,1;(h-127/r);",";(-87
/r+i)
5565 IF b=0 THEN  GO TO 5700
5570 FOR x=h-127/r TO h+128/r ST
EP o/r
5575 IF INKEY$="p" THEN  BEEP .3
,15: GO TO 5620
5580 LET z=(e+c*x)*(e+c*x)-4*b*(
f+d*x+k*x*x): IF z<0 THEN  NEXT
x: GO TO 5620
5590 LET g=(((-(e+c*x)+SQR z)/(2
*b))-i)*r+87: IF g>=0 AND g<=175
 THEN  PLOT ((x-h)*r+127),g
5600 LET y=(((-(e+c*x)-SQR z)/(2
*b))-i)*r+87: IF p$="s" THEN  GO
 SUB 5645
5605 IF y>=0 AND y<=175 THEN  PL
OT ((x-h)*r+127),y
5610 NEXT x
5620 PRINT #1;"   Pulsa tecla: M
 C A S": BEEP .5,20
5623 LET j$=INKEY$: IF j$="" THE
N  GO TO 5623
5625 IF j$="m" THEN  GO TO 101
5628 IF j$="c" THEN  COPY : GO T
O 101
5630 IF j$="a" THEN  BEEP .3,10:
 PAUSE 15: GO TO 5510
5635 IF j$="s" THEN  INPUT "Nomb
re? (10 letras)";n$: SAVE n$SCRE
EN$
5640 GO TO 5623
5645 LET s=-(r*SQR z)/b: IF g>17
5 AND y<0 THEN  PLOT ((x-h)*r+12
7),0: DRAW  OVER 1;0,175: RETURN

5646 IF y>175 AND g<0 THEN  PLOT
 ((x-h)*r+127),0: DRAW  OVER 1;0
,175: RETURN
5647 IF g>175 THEN  PLOT ((x-h)*
r+127),175: DRAW  OVER 1;0,-(175
-g-s): RETURN
5648 IF y>175 THEN  PLOT ((x-h)*
r+127),175: DRAW  OVER 1;0,(y-s-
175): RETURN
5649 IF g<0 THEN  PLOT ((x-h)*r+
127),0: DRAW  OVER 1;0,-(-s-g):
RETURN
5650 IF g<=175 AND y>=0 THEN  DR
```



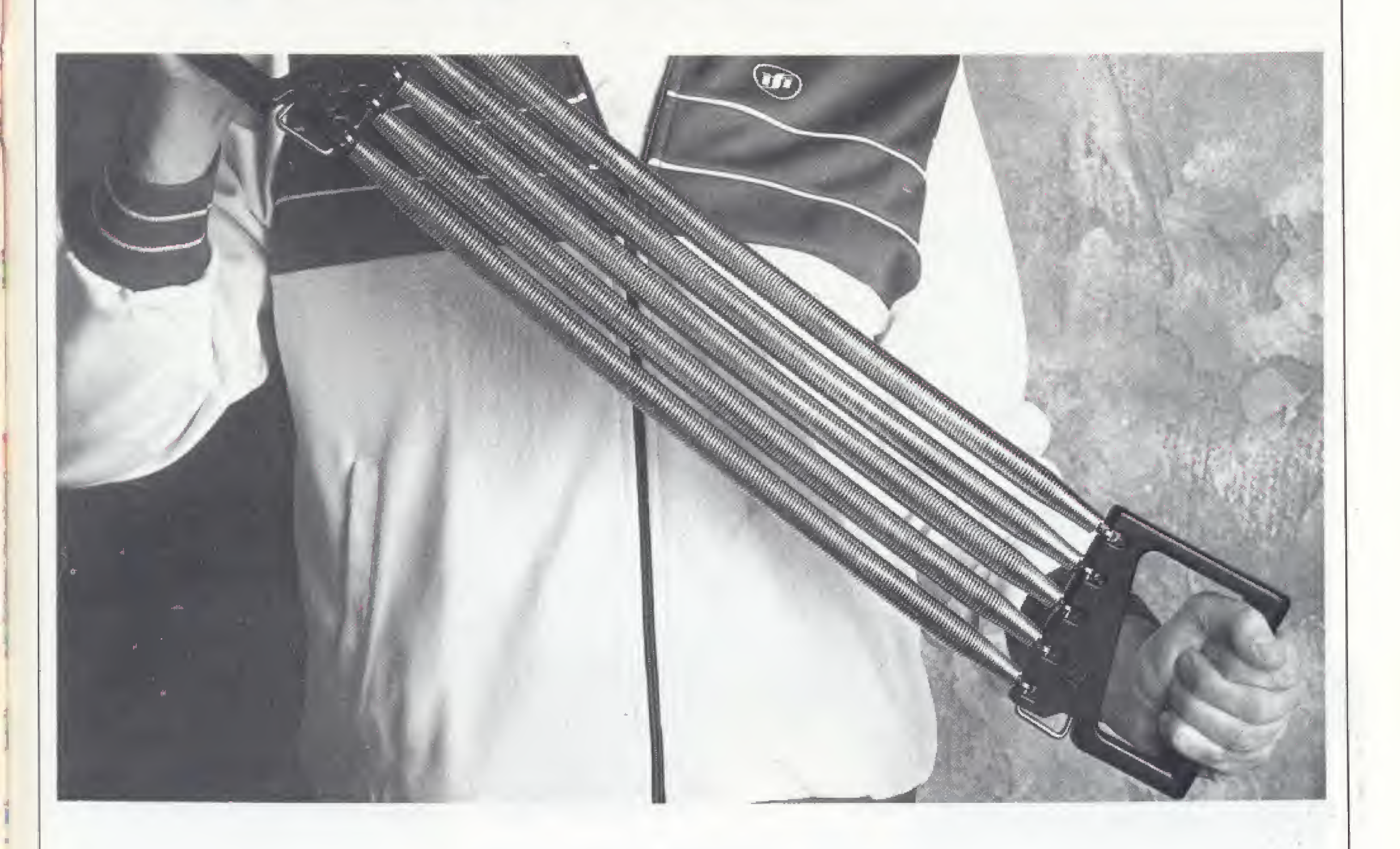
```
AW  OVER 1;0,s: RETURN
5655 IF y<0 THEN  PLOT ((x-h)*r+
127),0: DRAW  OVER 1;0,-(s-y): R
ETURN
5660 RETURN
5700 IF k=0 THEN  GO TO 5900
5705 FOR y=i-87/r TO i+88/r STEP
 o/r
5710 IF INKEY$="p" THEN  GO TO 5
620
5720 LET z=(d+c*y)*(d+c*y)-4*k*(
f+e*y): IF z<0 THEN  NEXT y: GO
TO 5620
5730 LET x=(((-(d+c*y)+SQR z)/(2
*k))-h)*r+127: IF x>=0 AND x<=25
5 THEN  PLOT x,((y-i)*r+87)
5740 LET g=(((-(d+c*y)-SQR z)/(2
*k))-h)*r+127: IF p$="s" THEN  G
O SUB 5760
5745 IF g>=0 AND g<=255 THEN  PL
OT g,((y-i)*r+87)
5750 NEXT y: GO TO 5620
5760 LET s=-(r*SQR z)/k: IF x>25
5 AND g<0 THEN  PLOT 0,((y-i)*r+
87): DRAW  OVER 1;255,0: RETURN
5763 IF g>255 AND x<0 THEN  PLOT
 0,((y-i)*r+87): DRAW  OVER 1;25
5,0: RETURN
5766 IF x>255 THEN  PLOT 255,((y
-i)*r+87): DRAW  OVER 1;-(255-x-
s),0: RETURN
5769 IF g<0 THEN  PLOT 0,((y-i)*
r+87): DRAW  OVER 1;-(s-g),0: RE
TURN
5772 IF g>255 THEN  PLOT 255,((y
-i)*r+87): DRAW  OVER 1;(g-s-255
),0: RETURN
5775 IF x<=255 AND g>=0 THEN  DR
AW  OVER 1;s,0: RETURN
5778 IF x<0 THEN  PLOT 0,((y-i)*
r+87): DRAW  OVER 1;-(-s-x),0: R
ETURN
5780 RETURN
5800 LET wv=4: GO TO 5400
5850 LET h=sx: LET i=sy: IF c=0
AND b=0 THEN  PLOT 127,0: DRAW 0
,175: GO TO 5550
5860 IF c=0 AND k=0 THEN  PLOT 0
,87: DRAW 255,0: GO TO 5550
5870 IF -(n+l*(h-127))/m-i+87>0
AND -(n+l*(h-127))/m-i+87<175 TH
EN  PLOT 0,-(n+l*(h-127))/m-i+87
: DRAW 255,255*(-l/m): GO TO 555
0
5880 PLOT -(n+m*(i-87))/l-h+127,
0: DRAW 175*(-m/l),175: GO TO 55
50
5900 FOR x=h-127/r TO h+128/r ST
EP o/r: IF INKEY$="p" THEN  GO T
O 5620
5905 IF (e+c*x)=0 THEN  NEXT x
5910 LET g=(-(f+d*x)/(e+c*x)-i)*
r+87: IF g>=0 AND g<=175 THEN  P
LOT ((x-h)*r+127),g: IF p$="s" A
ND g>=88 THEN  DRAW  OVER 1;0,17
5-g: NEXT x: GO TO 5620
5915 IF p$="s" AND g<88 AND g>=0
 THEN  DRAW  OVER 1;0,-g
5920 NEXT x: GO TO 5620
6000 REM CONICA POR CONDICIONES
6010 CLS : LET ww=0: DIM x(5): D
IM y(5)
6020 PRINT "COMO VAS A DEFINIR L
A CONICA ?  ",," 1-A PARTIR DE 5
 PUNTOS         ",," 2-1 TANGENT
E Y 4 PUNTOS "
6030 LET j$=INKEY$: IF j$="" THE
N  GO TO 6030
6040 GO TO 6030+(j$="1")*70+(j$=
"2")*170
6110 CLS : FOR n=1 TO 5: PRINT A
T 0,0;"Intoduce el punto ";n: IN
PUT "Punto (x,y)  x=";x(n);"  y=
";y(n): PRINT AT n,5;n;"(";x(n);
",";y(n);")": NEXT n
6120 RESTORE 6120: READ p,q,r,s,
t,u: DATA y(1)-y(2),x(2)-x(1),-p
*x(1)-q*y(1),y(3)-y(4),x(4)-x(3)
,-s*x(3)-t*y(3)
6130 READ k1,b1,c1,d1,e1,f1: DAT
A p*s,q*t,p*t+s*q,u*p+r*s,u*q+r*
t,u*r
6140 RESTORE 6140: READ p,q,r,s,
t,u: DATA y(1)-y(3),x(3)-x(1),-p
*x(1)-q*y(1),y(2)-y(4),x(4)-x(2)
,-s*x(2)-t*y(2)
6150 RESTORE 6130: READ k2,b2,c2
,d2,e2,f2
6160 LET p=k2*x(5)*x(5)+b2*y(5)*
y(5)+c2*x(5)*y(5)+d2*x(5)+e2*y(5
)+f2
6165 CLS : LET q=k1*x(5)*x(5)+b1
*y(5)*y(5)+c1*x(5)*y(5)+d1*x(5)+
e1*y(5)+f1: IF p=0 OR q=0 THEN
PRINT "   ESTOS PUNTOS ESTAN EN U
NA       CONICA DEGENERADA": GO
TO 101
```



```
6170 RESTORE 6170: READ s,k,b,c,
d,e,f: DATA -g/p,k1+k2*s,b1+b2*s
,c1+c2*s,d1+d2*s,e1+e2*s,f1+f2*s
6180 IF k<>0 THEN  RESTORE 6180:
 READ b,c,d,e,f,k: DATA b/k,c/k,
d/k,e/k,f/k,1: GO TO 6187
6182 IF b<>0 THEN  RESTORE 6182:
 READ k,c,d,e,f,b: DATA k/b,c/b,
d/b,e/b,f/b,1: GO TO 6187
6184 IF c<>0 THEN  RESTORE 6184:
 READ k,b,d,e,f,c: DATA k/c,b/c,
d/c,e/c,f/c,1
6187 IF ww=6 THEN  LET ww=0: GO
TO 1090
6188 PRINT " Estas condiciones d
efinen la   conica:",,k;" x^2 +"
,,b;" y^2 +",,c;" xy +",,d;" x +
",,e;" y +",,f;"  = 0": IF ww<>
5 AND ww<>4 THEN  PRINT ,,: PRIN
T  FLASH 1;"           PULSA UNA TE
CLA          ": PAUSE 30: PAUSE 0:
 GO TO 1090
6190 IF ww=5 THEN  LET ww=4: LET
 s=q: PRINT " SEGUNDA": GO TO 64
70
6193 LET ww=6: PRINT : PRINT "Es
coge una: Pulsa 1 o 2"
6195 IF INKEY$="1" THEN  LET s=p
: RESTORE 6470: READ k,b,c,d,e,f
: GO TO 6180
6197 IF INKEY$="2" THEN  LET s=g
: RESTORE 6470: READ k,b,c,d,e,f
: GO TO 6180
6198 GO TO 6195
6210 CLS : FOR n=1 TO 4: PRINT A
T 0,0;"Intoduce el punto ";n: IN
PUT "Punto (x,y)  x=";x(n);"  y=
";y(n): PRINT AT n,5;n;"(";x(n);
",";y(n);")": NEXT n
6225 RESTORE 6120: READ p,q,r,s,
t,u: RESTORE 6130: READ k1,b1,c1
,d1,e1,f1
6230 RESTORE 6140: READ p,q,r,s,
t,u: RESTORE 6130: READ k2,b2,c2
,d2,e2,f2
6240 INPUT "Introduce la TANGENT
E:                  ax + by + c =0

                 a=";p;" b=";q;" c=
";r
6450 CLS : RESTORE 6490: READ s,
t,u,s,t,u,s,t,u,s,t,v,s,t,v,s,t,
v,s,t,w,s,t,w,s,t,w
6453 IF u=0 AND w<>0 AND v<>0 TH
EN  LET s=-v/w: GO TO 6470
6455 IF w=0 AND v<>0 THEN  LET s
=-u/v: GO TO 6470
6460 LET s=v*v-4*u*w: IF s<0 THE
N  PRINT "CONICA DEGENERADA": GO
 TO 101
6465 LET p=(-v+SQR s)/(2*w): LET
 q=(-v-SQR s)/(2*w): IF p=0 AND
q=0 THEN  PRINT " ESTAS CONDICIO
NES SON DE UNA          CONICA DEG
ENERADA": GO TO 101
6466 IF q=0 THEN  LET s=p: GO TO
 6470
6467 IF p=0 THEN  LET s=q: GO TO
 6470
6468 LET ww=5: LET s=p: PRINT "
    Hay dos conicas con estas
condiciones: PRIMERA"
6470 RESTORE 6470: READ k,b,c,d,
e,f: GO TO 6180: DATA k1+k2*s,b1
+b2*s,c1+c2*s,d1+d2*s,e1+e2*s,f1
+f2*s
6490 DATA r*k1*b1+p*c1*e1/4+q*d1
*c1/4,e1*k1*q/2+c1*c1*r/4+b1*d1*
p/2,r*(s-t),p*f1*b1+q*d1*e1/4+r*
e1*c1/4,p*e1*e1/4+c1*f1*q/2+b1*d
1*r/2,u+p*(s-t),q*f1*k1+r*d1*c1/
4+p*e1*d1/4,k1*e1*r/2+c1*f1*p/2+
q*d1*d1/4,u+q*(s-t)
6492 DATA r*(k1*b2+k2*b1)+p*(c1*
e2+c2*e1)/4+q*(d1*c2+d2*c1)/4,(e
1*k2+e2*k1)*q/2+c1*c2*r/2+(b1*d2
+b2*d1)*p/2,r*(s-t),p*(f1*b2+f2*
b1)+q*(d1*e2+d2*e1)/4+r*(e1*c2+e
2*c1)/4,p*e1*e2/2+(c1*f2+c2*f1)*
q/2+(b1*d2+b2*d1)*r/2,v+p*(s-t),
q*(f1*k2+f2*k1)+r*(d1*c2+d2*c1)/
4+p*(e1*d2+e2*d1)/4,(k1*e2+k2*e1
)*r/2+(c1*f2+c2*f1)*p/2+q*d2*d1/
2,v+q*(s-t)
6494 DATA r*k2*b2+p*c2*e2/4+q*d2
*c2/4,e2*k2*q/2+c2*c2*r/4+b2*d2*
p/2,r*(s-t),p*f2*b2+q*d2*e2/4+r*
e2*c2/4,p*e2*e2/4+c2*f2*q/2+b2*d
2*r/2,w+p*(s-t),g*f2*k2+r*d2*c2/
4+p*e2*d2/4,k2*e2*r/2+c2*f2*p/2+
q*d2*d2/4,w+q*(s-t)
```

TENSORES



Los iniciados en el cálculo tensorial encontrarán indudablemente interesante este programa. Trabaja con tensores de dimensiones hasta cuatro. Las matrices aparecen en pantalla según se introducen y existe la posibilidad de rectificar errores. El menú tiene dos opciones: cambio de base y cambio de especie del tensor. Todas las explicaciones necesarias van incluidas en el programa.

G. Candela

```
  20 CLEAR : INPUT "Dimensio
n del espacio ?(1-4)";d: IF d>4
OR d<1 THEN  GO TO 20
  30 DIM c(d,d): DIM g(d,d): GO
TO 65
  35 CLS : PRINT  FLASH 1;AT 0,6
;"MATRIZ  DE CAMBIO  C"
  40 FOR n=1 TO d: FOR m=1 TO d:
 PRINT AT 19,0;"Introduce el ele
mento c(";n;",";m;"):": INPUT c(
n,m): PRINT AT n*3+m-2,(m*7-6);c
(n,m): NEXT m: NEXT n
  50 INPUT "Hay errores ? (s/n)"
;q$: IF q$="s" THEN  INPUT "Elem
ento erroneo ?: i=";n;" j=";m: I
F n>=1 AND m>=1 AND n<=d AND m<=
d THEN : PRINT AT n*3+m-2,m*7-6;
 OVER 1;c(n,m): PRINT AT 19,0;"I
ntroduce el elemento c(";n;",";m
;"):": INPUT c(n,m): PRINT AT n*
3+m-2,(m*7-6);c(n,m): GO TO 50
  55 IF q$<>"s" THEN  GO TO 1300
  60 GO TO 50
  65 CLS : PRINT  FLASH 1;AT 0,5
;"MATRIZ  FUNDAMENTAL  G"
  70 FOR n=1 TO d: FOR m=1 TO d:
```

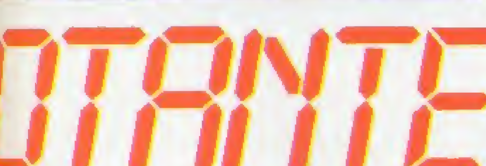
```
 PRINT AT 19,0;"Introduce el ele
mento g(";n;",";m;"):": INPUT g(
n,m): PRINT AT n*3+m-2,(m*7-6);g
(n,m): NEXT m: NEXT n
  80 INPUT "Hay errores ? (s/n)"
;q$: IF q$="s" THEN  INPUT "Elem
ento erroneo ?: i=";n;" j=";m: I
F n>=1 AND m>=1 AND n<=d AND m<=
d THEN : PRINT  OVER 1;AT n*3+m-
2,m*7-6;g(n,m): PRINT AT 19,0;"I
ntroduce el elemento g(";n;",";m
;"):": INPUT g(n,m): PRINT AT n*
3+m-2,(m*7-6);g(n,m): GO TO 80
  90 IF q$<>"s" THEN  GO TO 100
  95 GO TO 80
1000 CLS : PRINT "INTRODUCE LA E
SPECIE DEL TENSOR:
                     Indice de fil
a?                    1-COVARIANTE.
                      2-CONTRAVARIA
NTE."
1010 INPUT a: IF a<>1 AND a<>2 T
HEN  GO TO 1010
1015 PRINT AT 2,11;"columna?":
INPUT x: IF x<>1 AND x<>2 THEN
GO TO 1015
1020 PRINT AT 2,11;"fila de matr
iz?":  INPUT c: IF c<>1 AND c<>2
 THEN  GO TO 1020
1100 CLS : PRINT AT 0,10; FLASH
1;"TENSOR  T"
1105 DIM t(d,d,d): DIM r(d,d,d):
 FOR m=1 TO d
1110 FOR k=1 TO d: FOR n=1 TO d:
 PRINT AT 19,0;"Introduce el ele
mento t(";k;",";n;",";m;"):": IN
PUT t(k,n,m): PRINT AT k*3+n-2,(
n*7-6);t(k,n,m): NEXT n: NEXT k
1120 INPUT "Hay errores ? (s/n)"
;q$: IF q$="s" THEN  INPUT "Elem
ento erroneo ?: i=";k;" j=";n;"
k=";m: IF k>=1 AND k<=d AND n>=1
 AND m>=1 AND n<=d AND m<=d THEN
  PRINT AT k*3+n-2,(n*7-6); OVER
 1;t(k,n,m): PRINT AT 19,0;"Intr
oduce el elemento t(";k;",";n;",
";m;"):": INPUT t(k,n,m): PRINT
AT k*3+n-2,(n*7-6);t(k,n,m): GO
TO 1120
1130 IF q$<>"s" THEN  GO TO 1150
1140 GO TO 1120
1150 CLS : NEXT m
1160 GO TO 1250
1200 CLS : PRINT AT 5,2;"MENU:":
 PRINT : PRINT " 1-CAMBIO DE BAS
E.                2-CAMBIO DE ESP
ECIE."
1210 IF INKEY$="1" THEN  LET w=1
: LET g=0: GO TO 35
1215 IF INKEY$="2" THEN  GO TO 2
000
1220 GO TO 1210
1250 CLS : PRINT AT 11,11;"CALCU
LANDO": LET w=0: LET g=1: IF a>1
 THEN  GO TO 1350
1255 IF x>1 THEN  GO TO 1450
1260 IF c>1 THEN  GO TO 1550
1265 GO TO 1200
1350 CLS : PRINT AT 11,11; FLASH
 1;"CALCULANDO": FOR j=1 TO d: F
OR i=1 TO d: FOR l=1 TO d
1360 LET b=0: FOR m=1 TO d: IF w
=1 THEN  LET b=b+c(i,m)*t(m,l,j)
: NEXT m
1370 IF g=1 THEN  LET b=b+g(i,m)
*t(m,l,j): NEXT m
1390 LET r(i,l,j)=b
1400 NEXT l: NEXT i: NEXT j
1420 FOR j=1 TO d: FOR i=1 TO d:
 FOR l=1 TO d: LET t(l,i,j)=r(l,
i,j): NEXT l: NEXT i: NEXT j
1430 IF x=1 AND w<>1 THEN  GO TO
 1530
1450 FOR j=1 TO d: FOR i=1 TO d:
 FOR l=1 TO d
1455 LET b=0: FOR m=1 TO d
1465 IF w=1 THEN  LET b=b+t(i,m,
j)*c(l,m): NEXT m
1475 IF g=1 THEN  LET b=b+g(l,m)
*t(i,m,j): NEXT m
1490 LET r(i,l,j)=b
1500 NEXT l: NEXT i: NEXT j
1520 FOR j=1 TO d: FOR i=1 TO d:
 FOR l=1 TO d: LET t(l,i,j)=r(l,
i,j): NEXT l: NEXT i: NEXT j
1530 IF c=1 AND w<>1 THEN  GO TO
 1700
```

```
1550 FOR i=1 TO d: FOR l=1 TO d:
 FOR n=1 TO d: LET b=0: FOR m=1
TO d
1560 IF w=1 THEN  LET b=b+c(i,m)
*r(l,n,m): NEXT m
1565 IF g=1 THEN  LET b=b+g(i,m)
*r(l,n,m): NEXT m
1570 LET t(l,n,i)=b
1580 NEXT n: NEXT l: NEXT i
1585 IF g=1 THEN  LET g=0: GO TO
 1700
1600 DIM j(d,d): FOR j=1 TO d: F
OR i=1 TO d: LET b=0: FOR m=1 TO
 d: LET b=b+c(j,m)*g(m,i): NEXT
m: LET j(j,i)=b: NEXT i: NEXT j
1605 FOR j=1 TO d: FOR i=1 TO d:
 LET b=0: FOR m=1 TO d: LET b=b+
c(i,m)*j(j,m): NEXT m: LET g(j,i
)=b: NEXT i: NEXT j
1700 CLS : FOR j=1 TO d: PRINT "
TENSOR: MATRIZ ";j: FOR n=1 TO d
: FOR m=1 TO d
1710 PRINT AT n*3+m,m*7-6;t(n,m,
j)
1720 NEXT m: NEXT n: PRINT "Puls
a tecla": PAUSE 0: CLS : NEXT j
1730 CLS : PRINT "MATRIZ FUNDAME
NTAL G": FOR n=1 TO d: FOR m=1 T
O d: PRINT AT n*3+m,m*7-6;g(n,m)
: NEXT m: NEXT n: PRINT "Pulsa t
ecla": PAUSE 0: GO TO 1200
2000 REM CAMBIO DE ESPECIE
2010 CLS : PRINT "INTRODUCE LA N
UEVA ESPECIE :
                 Indice de fil
a?
                 1-COVARIANTE.

                 2-CONTRAVARIA
NTE."
2015 INPUT d: IF d<>1 AND d<>2 T
HEN  GO TO 2015
2020 PRINT AT 2,11;"columna?": I
NPUT e: IF e<>1 AND e<>2 THEN  G
O TO 2020
2025 PRINT AT 2,11;"fila de matr
iz?": INPUT f: IF f<>1 AND f<>2
THEN  GO TO 2025
```

NUMEROS EN COMA FL







Los números mayores que 65535, menores que −65535 y decimales se codifican en forma de coma flotante. En este artículo explicamos cómo se forman esos números para poder utilizarlos con el calculador.

En el Spectrum los números se pueden separar en 2 grandes grupos: los enteros del −65535 al 65535, a los que denominamos enteros normales, y los restantes números, llamados números en coma flotante.

ENTEROS NORMALES

Para ve más claro la codificación de estos números se debe ejecutar el programa 1. Para observar el resultado requerido se coloca un breakpoint en la línea 100 y se ve el contenido de los registros del Z-80.

La función de la subrutina STACK-FETCH es la de sacar del stack del calculador el quinto y ponerlo en el acumulador, el cuarto en el registro E, el tercero en el registro D, el segundo en el registro C, y el primero en el B.

Así pues, si has ejecutado el programa y has hecho el breakpoint, observarás que A=0, E=0, D=246d, C=9d y B=0. Los registros A y B en este tipo de codificaciones no son utilizados, el registro E al tener un 0 indica que el número es positivo, el C contiene el byte de menor peso y el C el de mayor; si el número fuera negativo el registro E contendría un 255d.

NÚMEROS EN COMA FLOTANTE

Para codificar un número en coma flotante se parte de un procedimiento matemático que es el siguiente: cualquier número, entero o decimal, se puede expresar como X por 2 elevado a Y, es decir, $X \cdot 2^Y$, donde X es la mantisa e Y el exponente.

Tras ejecutar el programa 2, y poner un breakpoint en la línea 100 observarás que al acumulador y el registro B, ya no son cero y que el registro E tiene un valor distinto de 0 o 255d, con lo que el número que estaba en el stack del calculador ya no es un entero normal, sino que está en coma flotante.

En el Spectrum los números se dividen en dos grandes grupos: los enteros normales y los números en coma flotante.

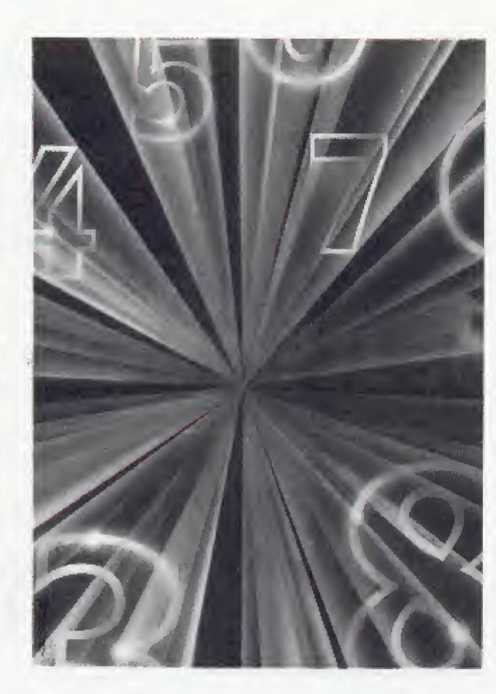
PROCEDIMIENTO DE CONVERSION A COMA FLOTANTE

Se divide en 2 partes fundamentales: determinación del exponente y determinación de la mantisa.

NUMEROS EN COMA FLOTANTE

Determinación del exponente

Tomamos el número 9.5, para codificarlo a coma flotante se halla un número entero n tal que 2 elevado a n sea inmediatamente mayor que 9.5. En este caso, 2 elevado a 3 es 8 con lo que 9.5 > 8, si n=4 entonces 9.5 < 16 con lo que el número entero buscado es 4. En el Spectrum el exponente se halla sumando 128 más n. Si has ejecutado el programa anterior verás que el acumulador tiene un 132, con lo que A contiene el exponente de la conversión a punto flotante.



Determinación de la mantisa

La mantisa está compuesta por 4 bytes. Siguiendo con el ejemplo anterior, se divide 9.5 entre 2 elevado a n y el resultado es 0.59375. Para hallar la mantisa se utiliza una serie numérica del tipo teniendo b valores entre 1 y 32 ya que la mantisa tiene 4 bytes (8x4=32).

Si b=1 el primer término de la serie será: $1/2^1 = Ø.5$, si el resultado obtenido anteriomente es mayor o igual que 0.5 lo restamos, así 0=59375–0.5=0.09375. Al restarlo ponemos a 1 el bit de mayor peso del primer byte. Como este bit siempre va a estar a 1 si las operaciones son correctas, el Spectrum lo usa para identificar si el número es positivo, bit a 0, o si es negativo, bit a 1.

La serie mencionada seguirá: $1/2^2$=Ø.125, 0.0625, 0.03125, 0.015625, hasta b=32 inclusive. Para b=2 como 0.09375 es menor que 0.25 será un bit puesto a cero, para b=3 0.09375 sigue siendo menor con lo que el siguiente bit será otro cero, pero para b=4 0.09375 es mayor que 0.0625 con lo que se lo restamos, 0.09375–0.0625=0.03125 y ponemos a 1 el bit de la sucesión b=4. Para b=5, al ser el resultado de la resta anterior igual al valor de la sucesión para b=5, lo restamos y ponemos el bit b=5 a 1. Como el resultado es 0, todos los demás bits desde b=6 hasta b=32 serán cero, con lo que la conversión ha terminado.

CONTENIDO DE LOS REGISTROS

Después de explicar cómo se halla la mantisa y el exponente, observarás que el acumulador contiene un 132, que E, que corresponde al primer byte de la mantisa, tendrá un valor 24d, D, que es el segundo byte, C, el tercero, y B, el cuarto son todos cero.

Como sabemos que el proceso de conversión es lento y complicado, hemos realizado un programa BASIC que convierte cualquier número a coma flotante (Listado 6).

LISTADO 1

```
  10         ORG   50000
  20         LD    A,255
  30         CALL  #2D28;STK-A
  40         LD    A,10
  50         CALL  #2D28;STK-A
  60         RST   #28;10,255
  70         DEFB  #04;255*10
  80         DEFB  #38;2550
  90         CALL  #2BF1;STK-FETL
H
 100         RET
```

LISTADO 2

```
  10         ORG   50000
  20         LD    A,19
  30         CALL  #2028;STK-A
  40         LD    A,2
  50         CALL  #2028;STK-A
  60         RST   #28;2,19
  70         DEFB  #05;19/2
  80         DEFB  #38;9.5
  90         CALL  #2BF1;STK-FETC
H
 100         RET
```

LISTADO 3

```
  10         ORG   50000
  20         RST   #28
  30         DEFB  #34;STK-DATA
  40         DEFB  243;EXPONENTE
      MANTISA
  50         DEFB  %01111011
  60         DEFB  %00110011
  70         DEFB  %00110011
  80         DEFB  %00110011
  90         DEFB  #38
 100         CALL  #2DE3;PRINTFP
 110         RET
```

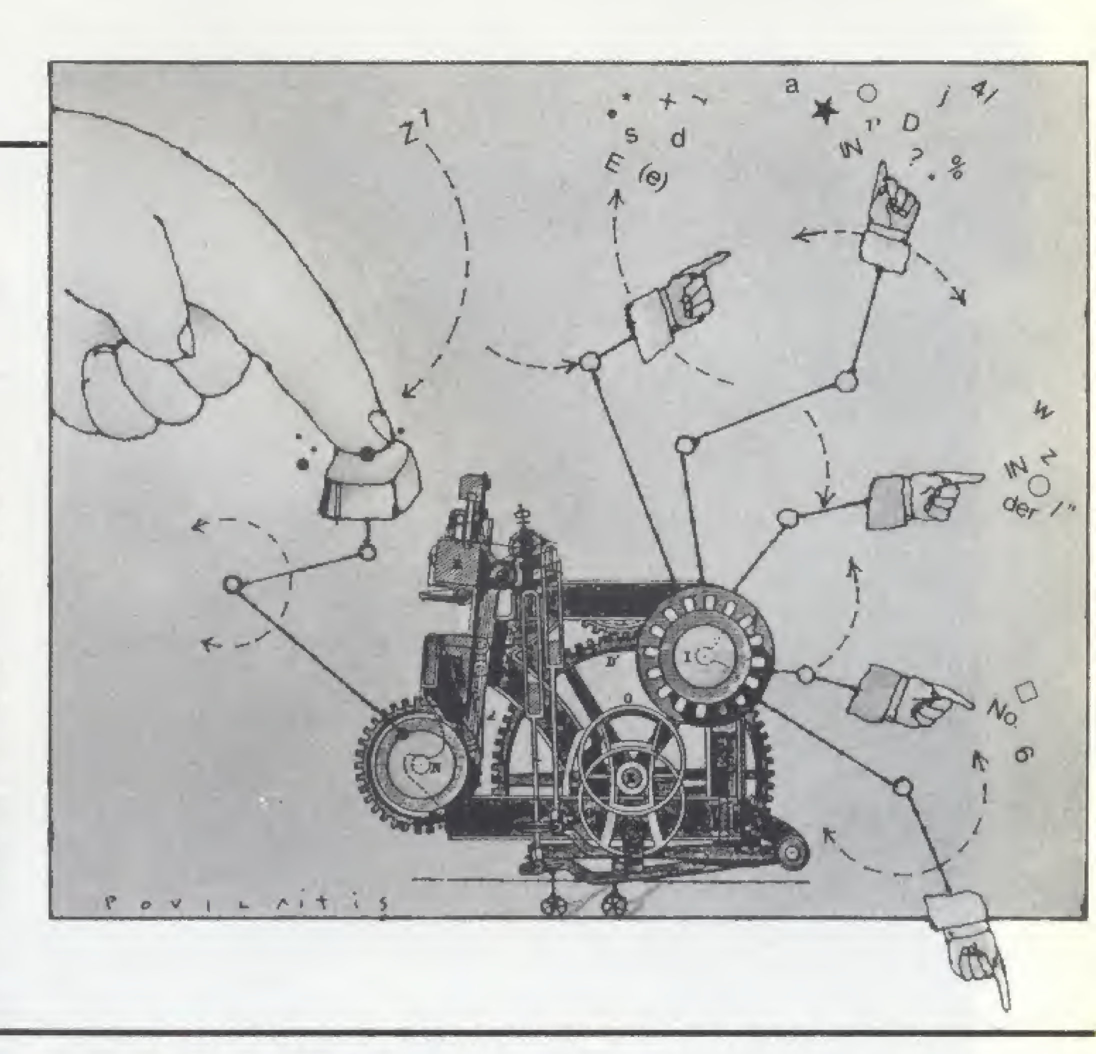
LISTADO 4

```
  10         ORG   50000
  20         RST   #28
  30         DEFB  #34;STK-DATA
  40         DEFB  52;EXPONENTE
      MANTISA
  50         DEFB  24 ;0,0,0
  60         DEFB  #38
  70         CALL  #2DE3;PRINTFP
  80         RET
```

LISTADO 5

```
  10         ORG   50000
  20         LD    A,132;EXPONENT
E
  30         LD    E,24;MANTISA
  40         LD    D,0;      "
  50         LD    C,0;      "
  60         LD    B,0;      "
  70         CALL  #2AB2;LOS REGI
STROS DEL CALCULADOR
  80         CALL  #2DE3;PRINTFP
  90         RET
```

COMA FLOTANTE EN FORMA DE STK-DATA



La rutina llamada stk-data, cuyo código del calculador es 34h (52d), permite introducir en el stack del calculador cualquier número, pero en coma flotante y de una manera algo peculiar.

Un número en coma flotante usa 5 bytes para su codificación. En la forma stack data puede utilizar 2, 3, 4 o 5 bytes para ese mismo número. En esta forma, el exponente especifica el número de bytes que necesita la mantisa para determinar el número y además, el exponente real. Hay que dejar claro que si el exponente indica que el número de bytes nece-

sarios para la mantisa es 2, los 2 restantes contienen cero.

El mismo programa BASIC, convertidor a coma flotante, transforma el número a la forma de stack data. El programa parte de un número es coma flotante, lo transforma al tipo stack data. Vamos a explicar ahora el proceso inverso: como partiendo de la forma stack data, al Spectrum sabe cuál es el exponente real y el número de bytes de la mantisa.

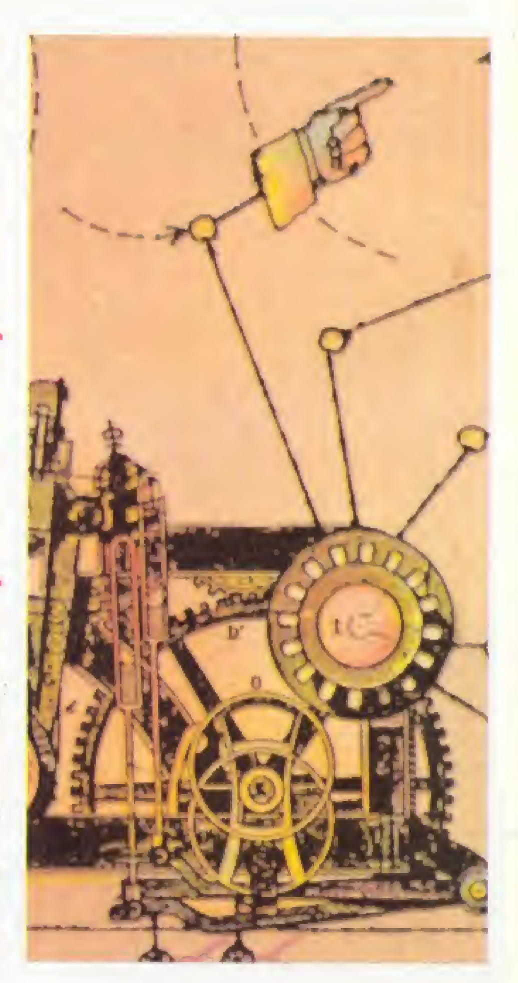
El primer byte que sigue al código 34h (52d) es el exponente en forma stack data. Para hallar el exponente real divide el exponente en forma stack dta entre 64d. El entero del cociente más 1 indica el número de bytes de la mantisa que tiene ese número en la forma stack data. El resto de la anterior división más 80d y da el exponente real.

Unos ejemplos de este procedimiento están expuestos en los listados 3 y 4. En el listado 4 el número a introducir es el 7.85 y en el 5 el 9.5. Aunque éste es el sistema utilizado en el Spectrum para introducir números en el stack del calculador, también se pueden introducir en la forma normal en coma flotante (listado 5).

Antonio Luis Matías y Francisco Javier Rodríguez



La rutina STK-DATA, cuyo código del calculador es 34 le permite introducir en el stack del calculador cualquier número en una forma algo peculiar.



LISTADO 6

```
 10 LET COC=0: LET M=0: LET X=5
: INPUT "NUMERO:";NU: LET NU=ABS
 NU
 90 PRINT " NUMERO  ";NU: PRINT

100 FOR I=-128 TO 128
110 IF NU<2^I THEN  GO TO 200
120 NEXT I
200 PRINT "EXPONENTE= ";128+I
210 LET Q=NU/2^I
220 FOR B=1 TO 32
230 IF Q>=1/2^B THEN  GO TO 300
240 PRINT 0;: LET M=M+1
250 IF B=8 OR N=16 OR B=24 THEN
 PRINT
255 NEXT B
260 IF B>32 THEN  GO TO 400
300 PRINT 1;: LET M=M+1: IF B=8
OR B=16 OR B=24 THEN  PRINT
305 IF B>=9 AND B<=16 THEN  LET
 COC=1
306 IF B>=17 AND B<=24 THEN  LE
T COC=2
307 IF B>=25 AND B<=33 THEN  LE
T COC=3
310 LET Q=Q-1/2^B
320 GO TO 255
400 PRINT AT 9,0;"RECUERDA QUE
EL BIT DE MAYOR    PESO DEL PRIM
ER BYTE INDICA EL  SIGNO"
410 LET E=128+I
420 LET R=E-80
430 LET DAT=(64*COC)+R
435 PRINT : PRINT

440 PRINT "EXPONENTE EN STK-DAT
A  ";DAT
445 PRINT
450 PRINT "NUMERO DE BYTE ";COC
+1
```

infodis, s.a. LE OFRECE LOS MEJORES LIBROS PARA SU ORDENADOR



P.V.P. 750 PTAS.
(IVA INCLUIDO)
Descubre los misterios de la programación de una forma sencilla, con ejemplos, programas y organigramas.
(110 páginas, tamaño 13,5 x 21)



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Con utilidades, juegos explosivos y gráficos dinámicos que lleva al BASIC hasta el mejor aprovechamiento de sus posibilidades.
(200 páginas, tamaño 15,5 x 21,5).



P.V.P. 750 PTAS.
(IVA INCLUIDO)
Un libro especialmente dedicado a los que se inician por vez primera en el mundo del Spectrum.
(100 páginas, tamaño 13,5 x 21).



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Una inestimable ayuda que complementará la que proporciona el manual del ordenador.
(108 páginas tamaño 13,5 x 21,5).



P.V.P. 900 PTAS.
(IVA INCLUIDO)
Un compendio de los programas más diversos con los que podrá aprender jugando las importantnes características del BASIC.
(258 páginas, tamaño 15,5 x 21,5).



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Muestra una visión más completa del correcto funcionamiento del juego de instrucciones del C-64.
(108 páginas, tamaño 13,5 x 21,5).

CUPON DE PEDIDO

enviar a:

infodis, s.a.

C/BRAVO MURILLO, 377
28020 MADRID

COPIE O RECORTE ESTE BOLETIN DE PEDIDO.

DESEO RECIBIR LOS SIGUIENTES TITULOS:

15 HORAS CON EL SPECTRUM (P.V.P. 750) ☐
LOS MEJORES PROGRAMAS PARA EL ZX SPECTRUM (P.V.P. 900) ☐
LOS MEJORES PROGRAMAS PARA EL COMMODORE 64 (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL I (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL II (P.V.P. 800) ☐
(más 100 ptas. de gastos de envio).

El importe lo abonaré POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO ☐ American Express ☐ Visa ☐ Interbank ☐

Número de mi tarjeta: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

NOMBRE ____________________

CALLE ____________________

CIUDAD ____________________

PROVINCIA ____________________ C. P. __________



APARTADO DE CORREOS

Dirige tus cartas a:
Todospectrum
Bravo Murillo, 377, 5.º-A
28020 Madrid

OPERATIVO VOLATIL

En un número anterior de su estupenda revista apareció un artículo con el título UN NUEVO SISTEMA OPERATIVO PARA EL SPECTRUM, realizado el programa propuesto, aparecieron las deseadas 64 columnas y todos los comandos adicionales (el programa es estupendo).

Con este OPERATIVO comencé un programa propio que, después de pulirlo y perfeccionarlo, funcionó perfectamente. Después del SAVE correspondiente desconecté el ordenador y... al día siguiente comenzaron mis problemas, pues al cargarse el programa este funcionó ¡en 32 columanas y no en 64 como fue realizado!

Mi pregunta concreta es cómo puedo hacer para que mis programas salgan por pantalla en 64 columnas.

Vicente Serrador
Mislata (Valencia)

El nuevo sistema operativo que implementa este programa no es más que una copia en RAM del bucle principal del operativo del Spectrum, a la que se han hecho una serie de modificaciones y se le han añadido algunos comandos y funciones nuevas. Por lo tanto este programa deberá estar en memoria ejecutándose si pretendes hacer uso de alguna de sus ventajas. No basta con que hagas un programa cuando esté vigente este operativo para que vuelva a aparecer cada vez que lo cargues, ya que, como sabrás, la memoria RAM de tu ordenador es volátil, lo que quiere decir que los datos que se encuentren en ella sólo se conservarán mientras el ordenador permanezca encendido. Deberás, pues, incluir al principio de tu programa una serie de instrucciones que carguen y «lancen» el nuevo operativo, y salvar tras aquél el código correspondiente.

LOBOTOMIA PARA EL SPECTRUM

— ¿Es posible substituir algunas pastillas de la ROM por unas RAM, teniendo en cuenta el cambio de las otras ROM por unas EPROM de programa totalmente diferente a la ROM standard del Spectrum? (Naturalmente, estas RAM deberían ser POKEables en C/M).

— En caso de que se pudieran cambiar, ¿han de ser TMS 4532 (L3/L4) o es posible utilizar otras?

— El interface TV PAL que envía la señal de TV en el Spectrum obtiene la información de las direcciones de memoria 4000H-5800H, parece ser que esto es independiente del programa en ROM. ¿Es posible cambiar las direcciones de archivo de pantalla por medio de Hardware?

— También parece ser que el Maskable Interrupt sucede periódicamente, e independientemente del programa ROM se dirige a la Restart Routine 0038H, ¿es esto cierto?

David Sáez
Barcelona

En teoría es posible cambiar parte de la ROM por RAM, pero has de saber que los 16 Kbytes de ROM se encuentran en un único chip, que habría que sustituir por una EPROM y varios de RAM. En cuanto a los chip a utilizar, difícilmente podrían ser los 4532 de Texas Instruments, pues es muy raro el encontrarlos a la venta en comercios; además, el utilizar cualquier chip de RAM dinámica daría ciertos problemas para conseguir que se efectuara el «refresco» con normalidad, por lo que parece más aconsejable usar alguno de RAM estática.

El encargado de leer la memoria de pantalla para envíar al modulador la información necesaria es la polifacética ULA, por lo que cambiar las direcciones de esta zona es inviable, al menos desde el punto de vista práctico. Además, el uso que se hace en el Spectrum de un doble bus de datos para esta zona haría necesario situarla en esas 16 primeras Ks de RAM.

En cuanto a las interrupciones enmascarables, en el modo 1 se salta inevitablemente a la dirección que señalas, pero pueden prohibirse las interrupciones con DI o usar el modo 2 para que salte a la dirección que nos interese. Para esto sería necesario dar al registro I el valor apropiado para que apunte a la dirección en que, a su vez, vamos a almacenar la dirección de la rutina de interrupciones, y, tras esto, hacer IM 2 antes de nada en el punto de entrada de la ROM, allí donde se inicializa el sistema (dirección 0000H). En cuanto a las interrupciones no enmascarables (NMI), saltan siempre a la dirección 0066H, donde deberá estar su correspondiente rutina (en el Spectrum estándar, por un error en esta rutina, las NMI no pueden ser utilizadas).

Como verás, no es nada fácil intentar cambiar el «carácter» al Spectrum de esta manera, hacen falta buenos conocimientos de la arquitectura de la máquina y un tanto de programación y electrónica aplicada. Para el caso de que te decidas a aventurarte en este sentido te deseamos mucha suerte. Esperamos tus noticias.

COLABORADOR INTERESADO

Quisiera que me informaran sobre las siguientes cuestiones: ¿pagan por los programas que mandan los lectores? ¿qué condiciones ponen a los programas? ¿cuánto dinero dan por programa?

David Castro
La Laguna (Tenerife)

Por supuesto que pagamos los programas o artículos que publicamos procedentes de lectores de toda España. Los programas deben estar grabados en cassette o microdrive, con texto explicativo de su utilidad y modo de empleo, y los artículos mecanografiados a doble espacio. A efectos de remuneración se analiza cada colaboración aisladamente, estudiando su calidad y complejidad. ¡No te hagas esperar!

RARILLO EL QL

Soy poseedor de un QL desde hace ya seis meses y he de decir que estoy muy contento con él. Sin embargo he notado una serie de curiosidades en su SuperBASIC que me gustaría comentaros.

1) Según el manual de QL, la instrucción CLS admite dos parámetros: el número de canal afectado (o el de omisión), y la parte de la pantalla que se ha de borrar (toda en la omisión). Mi duda es repecto al segundo parámetro; según el manual este puede ser 0, 1, 2, 3 0 4; que borran toda, la parte superior, la inferior, la línea del cursor y desde el cursor al final de la línea respectivamente.

Si bien esto es cierto, no es toda la verdad, pues este parámetro puede tomar muchos otros valores de la gama entre −32767 a 32767, si bien no todos, pues muchos dan el error «Parámetro incorrecto». El efecto de estos valores es de los más diverso, desde simplemente borar parte de la pantalla, hasta hacer un scrolling y un panning parcial de la misma, pasando por cambios de color del papel y la tinta, del flash, del subrallado, y algunos efectos más curiosos como la activación del cursor en el canal especificado o un crash que bloquea todo el sistema (con el consabido reset para solucionarlo).

Mi pregunta es la siguiente: ¿se trata esto de una errata del sistema o tiene un utilidad determinada? Sí, es así, ¿cuál es esta y cómo se utiliza?

Juan I. Rodríguez
Castilleja de la Cuesta (Sevilla)

Contestamos a tus preguntas en el orden que las has formulado:

La instrucción CLS puede, efectivamente, tener como segundo parámetro un número entre 0 y 4, con las funciones que has comentado. Si ponemos aquí otra cifra cualquiera podemos llegar a engañar a la máquina de modo que acabe haciéndose un lío. Esto no es en realidad una errata, pues, como bien se menciona en el manual de instrucciones, esa instrucción no debería usarse con otros valores que los indicados. Bien es cierto que un lenguaje como el BASIC debe preveer todas las equivocaciones que pueda tener el programdor para dar el correspondiente mensaje de error, y quizá sea esta una de las causas de que este lenguaje se haya hecho tan popular, pero no es menos cierto que el SuperBASIC no es un BASIC cualquiera, en él se ha sacrificado en parte de esa forma de ser del BASIC para hacerlo más potente y versátil. En todo caso, agradecemos que nos hayas comunicado tus experiencias en este sentido y esperamos lo sigas haciendo.

EL CORCHO

Se vende Seikosha GP-500A en su embalaje original, con menos de medio año de uso (matriz 7*5, 80 columnas, 50 cps y dos tipos de escritura) con manual, software (LLIST LPRINT, COPY y Copy de grises) e interface de Pin Soft con cable. Precio a convenir al Tel.: 246 07 61 de Barcelona. Preguntar por David o por Daniel.

Intercambio programas para Spectrum. Me interesa conseguir el compilador COLT y el ensamblador GENS. José Luis López López. Palamios, 5, 2.° D. 36001 Pontevedra.

Vendo Spectrum 48K completo, magnetofon Sonisur, y demás elementos. Todo 30.000 ptas. (regalo juegos y revistas), llamar a Enrique Martos. Lepanto, 4, 1.° 2. Tel.: 34 00 96. 34006 Málaga.

Vendo Spectrum 48K, con impresora Seikosha GP-50 S, joystick, interface Kempston y accesorios. Además incluyo revistas, programas de juegos y utilidades y manuales de Basic. Precio razonable. Robert Gamigos. Pje. St. Ramón Nonato, 3, 1.° 2a. 08028 Barcelona. Tel.: (93) 249 49 85.

Vendo barato Spectrum 48K con fuente de alimentación, cables y varios en tan sólo 20.000 ptas. Regalo con ello revistas de Todospectrum y libros en Código Máquina. Alfonso Mier Herreria. Sen Fernando, 2, 4.° A. Sta. Cruz de Benzana (Cantabria).

Teclados de repuesto. Dispongo de teclados originales completos (teclas, membrana y carcasa) por 3.000 ptas. Puedo llevarlos hasta Madrid. Alfredo Villalba. Tel.: (911) 22 68 06. Guadalajara.